**ЛАБОРАТОРНА РОБОТА №4**

**КОЛІРНА ОБРОБКА ІНФОРМАЦІЇ**

**Колірні моделі**

Призначення колірної моделі полягає в тому, щоб зробити можливим опис кольорів певним стандартним, загальноприйнятим способом. По суті, колірна модель визначає деяку систему координат і підпростір всередині цієї системи, в якому кожен колір представляється єдиною точкою.

Більшість сучасних колірних моделей орієнтовані або на пристрої відтворення кольору (наприклад, кольорові монітори, принтери тощо), або на певні прикладні завдання (такі як створення кольорової графіки в анімації), коли робота з кольором являється безпосередньою ціллю. Апаратно-орієнтованими колірними моделями, найбільш часто використовуваними на практиці, являються модель RGB для кольорових моніторів і широкого класу кольорових відеокамер, моделі CMY і CMYK для кольорових принтерів і модель HSI, яка добре відповідає кольоросприйняттю людини. Остання модель має також ту перевагу, що вона розділяє колірну і яскравну інформацію на зображенні і тому дає можливість застосовувати багато напівтонових методів обробки зображень. В даний час використовується безліч різних колірних моделей; це обумовлено тим, що наука про колір є широкою областю, яка включає в себе численні програми.

* + 1. **Колірна модель RGB**

B RGB моделі кожен колір представляється червоним, зеленим і синім первинними основними кольорами (компонентами). В основі моделі лежить декартова система координат. Кольорова палітра є кубом. Точки, які відповідають червоному, зеленому і синьому кольорам, розташовані в трьох вершинах куба, що лежать на координатних осях. Блакитний, пурпурний і жовтий кольори розташовані в трьох інших вершинах куба. Чорний колір знаходиться на початку координат, а білий - в найбільш віддаленій від початку координат вершині. У моделі відтінки сірого кольору (точки з рівними RGB значеннями) лежать на діагоналі, що з'єднує чорну і білу вершини. Різні кольори в цій моделі представляють собою точки на поверхні або всередині куба і визначаються вектором, проведеним в дану точку з початку координат. Для зручності передбачається, що всі значення кольору нормовані таким чином, щоб куб був одиничним кубом, тобто все значення R, G і В лежать в діапазоні [0,1]. Подані в колірній моделі RGB зображення складаються з трьох окремих зображень-компонент, по одному для кожного первинного основного кольору. При відтворенні RGB монітором ці три зображення змішуються на люмінісцентному екрані і утворюють складене кольорове зображення. Число бітів, використовуваних для подання кожного пікселя в RGB просторі, називається глибиною кольору. Розглянемо RGB зображення, в якому

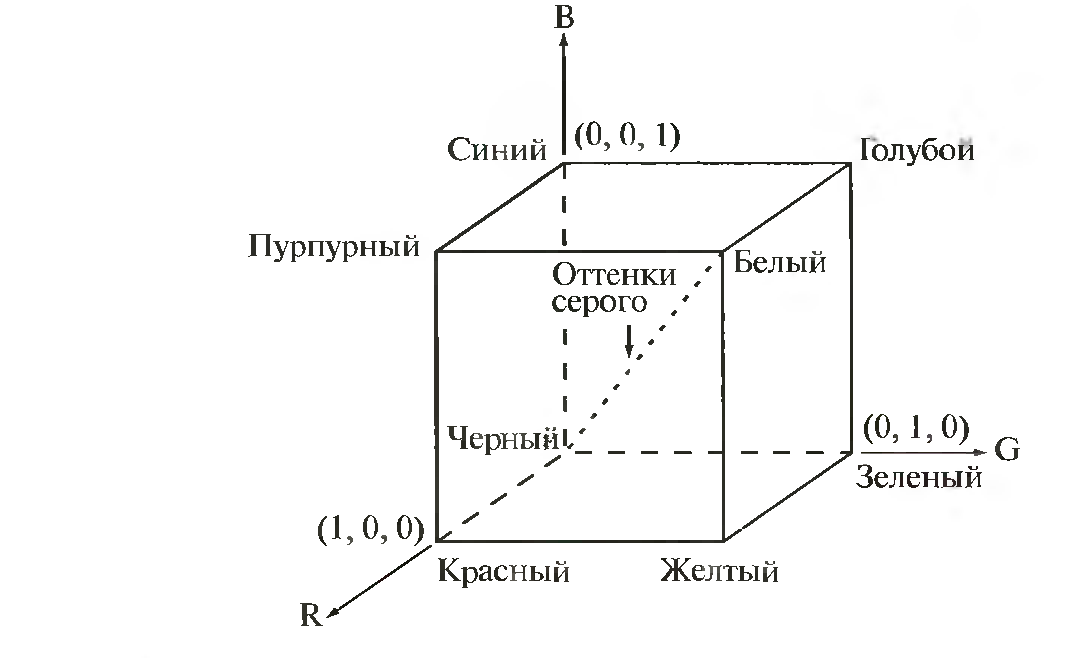
кожна з компонент - червона, зелена і синя - є 8-бітовою. У такому випадку говорять, що кожен кольоровий RGB піксель (тобто триплет значень (R, G, B)) має глибину 24 біта (три колірні площини помножити на число бітів в кожній площині); для такого зображення часто використовується термін повноколірне зображення. Сумарне число можливих кольорів в 24-бітовому RGB зображенні становить 16.777.216.

Хоча, високоякісні графічні адаптери і монітори забезпечують хороше відтворення кольорів 24-бітових RGB зображень, багато систем, які використовуються в даний час є обмежені кількістю кольорів, рівним 256. Крім того, існує цілий ряд додатків, в яких має сенс використовувати не більше сотні, а то і меншу кількість кольорів. Хорошим прикладом тут можуть служити методи обробки зображень в псевдокольорах. Також бажано мати підмножина кольорів, які б відтворювалися точно у всіх використовуваних графічних системах незалежно від їх особливостей. Така підмножина кольорів називається палітрою фіксованих RGB кольорів або набором кольорів, однаково відтворюваних усіма системами. Стосовно інтернет-додатків ця підмножина кольорів називається палітрою фіксованих Web кольорів або набо-

ром кольорів, однаково відтворюваних усіма програмами перегляду інтернет-сайтів.

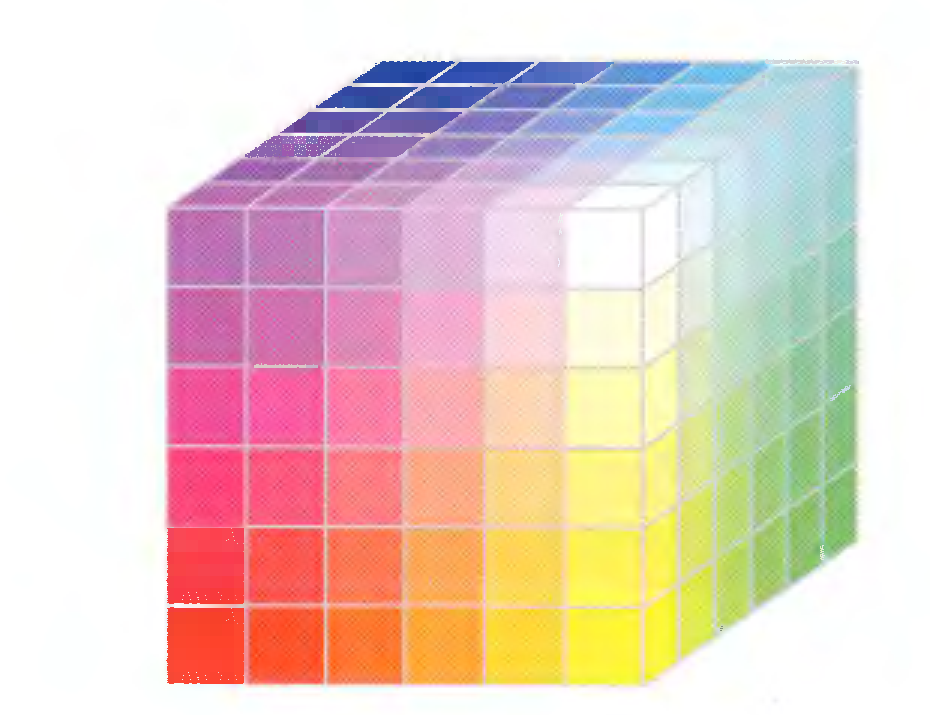
Виходячи з твердження, що 256 кольорів - це той мінімальний набір кольорів, які точно відтворюються будь-яким графічним пристроєм, то корисно мати загальноприйнятий стандарт запису цих кольорів. Відомо, що сорок з цих 256 кольорів відтворюються різними операційнимит системами по-різному; при цьому залишається 216 кольорів, які є загальними для більшості систем. Ці 216 кольорів стали стандартом фіксованих кольорів, особливо для інтернет-додатків.

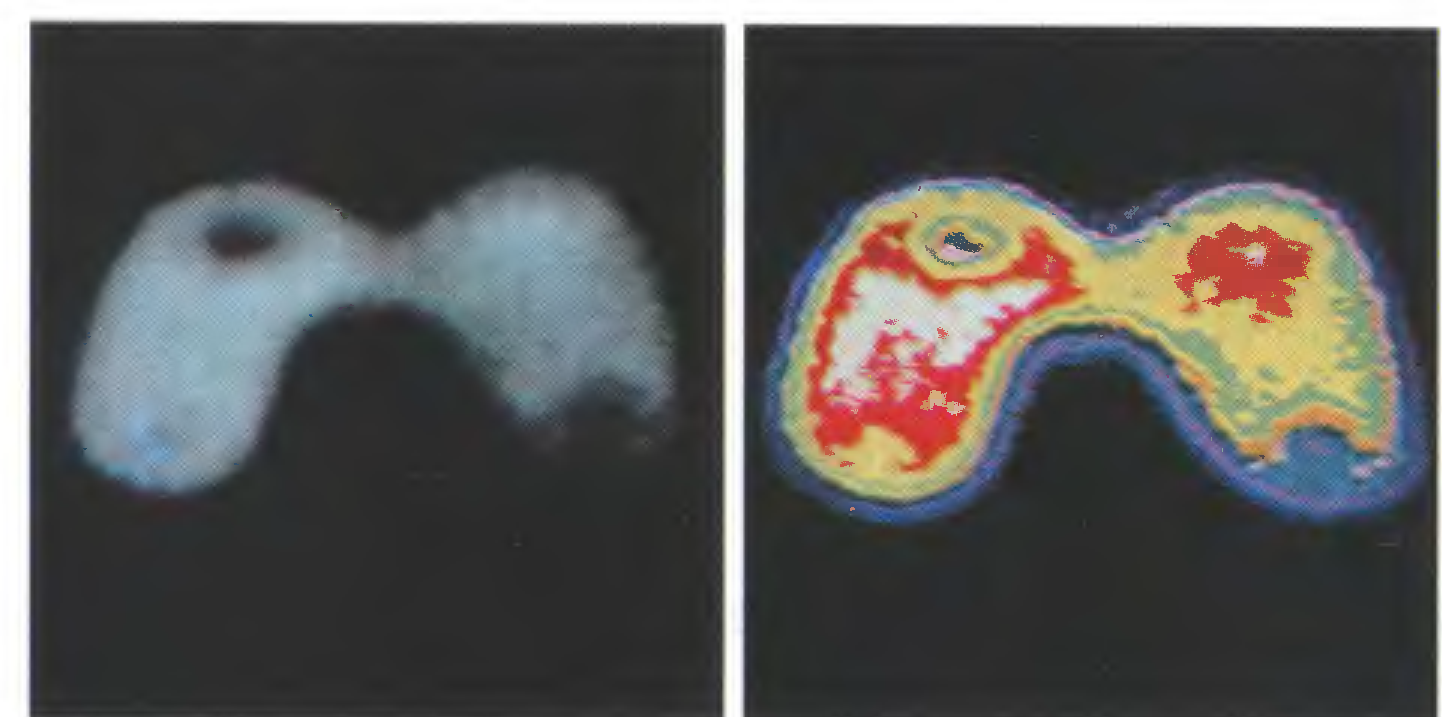
Оскільки для формування RGB кольору потрібно три числа, кожен колір з палітри фіксованих кольорів задається трьома двозначними шістнадцятковими числами. Наприклад, чистому червоному кольору відповідає FF0000. Значення 000000 і FFFFFF відповідають чорному і білому кольорам, відповідно. Ті ж результати виходять, звичайно, і при використанні більш звичної десяткової системи числення. Наприклад, найяскравіший червоний колір має компоненти R = 255 (FF) і G = B = 0.







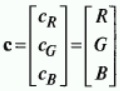




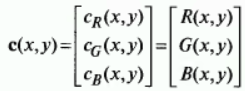
**1.2. Основи обробки кольорових зображень**

Використовувані при обробці кольорових зображень підходи поділяються на дві основні категорії. Підходи першої категорії вважають, що кожен кольоровий компонент зображення обробляється окремо, а потім результуюче кольорове зображення складається з відповідних частинок, оброблених окремо. Для підходів другої категорії характерна безпосередня робота з кольоровими пікселями. Оскільки кольорове зображення вміщає, як мінімум, три складові частини, то значення пікселя зображення представляє собою вектор. В RGB моделі, наприклад, значення пікселя зображення може розглядатися як вектор, проведений з початку координат в відповідну точку колірного простору.

Нехай **c** є довільний вектор в колірному просторі RGB:



Цей вираз показує, що компонентами вектора **c** є RGB координати точки в колірному просторі. Запис



Відображає той факт, що компоненти вектора **c** залежить від просторових координат (x,y). Для зображення розмірами MxN існують MN таких векторів **c**(x,y), x=0, 1, 2, …, M-1, y=0, 1, 2, …, N-1.

Важливо розуміти те, що другий вираз описує вектор, компоненти якого є функціями просторових змінних x і у. Ця обставина часто є джерелом непорозуміння, яких можна уникнути, якщо сконцентруватися на тому, що наш інтерес лежить в області просторових методів обробки. А саме, нас цікавлять неодноріні методи обробки зображень, тобто залежні від просторових координат х і у. Той факт, що пікселі зображення є кольоровими, призводить в найпростішому випадку до того, що ми маємо можливість обробляти незалежно кожну кольорову компоненту зображення, використовуючи звичайні методи обробки напівтонових зображень. Однак отримані таким способом результати не завжди збігаються з результатами обробки, виконуваної безпосередньо в колірному векторному просторі; в цьому випадку потрібна розробка нових підходів.

Для того, щоб методи покомпонентной обробки і векторної обробки були еквівалентні, необхідне виконання двох умов. По-перше, метод повинен бути застосовним як до векторів, так і до скалярів. По-друге операції над кожною компонентою вектора не повинні залежати від інших компонент. Але кожна компонента усередненого вектора дорівнює середньому по околиці від значень зображення цієї компоненти. Тому, той самий результат буде отримано, якщо виконати усереднення окремо, по ​​кожної компоненті і потім з усереднених компонент сформувати шуканий вектор.

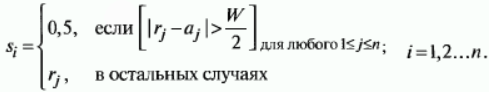
**1.3. Перетворення кольорів**

Перетворення кольорів, являє собою обробку компонент кольорового зображення в межах однієї, окремо взятої колірної моделі. Це відрізняє дані перетворення від перетворень координат кольору при переході з однієї колірної моделі до іншої.

**1.3.1. Вирізання діапазону кольорів**

Вирізання певного діапазону кольорів на зображенні використовується для виділення певних об’єктів з їхнього оточення. Основна ідея заключається в тому, щоб або (1) відтворити кольори, що нас цікавлять так, щоб вони виділялися на загальному фоні, або (2) використовувати визначаються кольором області в якості маски при подальшій обробці. Найбільш простий підхід полягає в тому, щоб узагальнити метод вирізування діапазону кольорів. Однак оскільки значення кольорового пікселя є n -мірний вектор, функції перетворення для вирізання діапазону кольорів є більш складними, ніж аналогічні функції для вирізання діапазону яскравості. Це пов'язано з тим, що при будь-якому відомому на практиці метод вирізування колірного діапазону кожна колірна компонента перетворюваного пікселя залежить від усіх n-колірних компонент початкового пікселя.

Один з найпростіших способів «розділити» кольорове зображення полягає в тому. щоб відобразити всі кольори. що лежать поза областю інтересу, в деякий нейтральний, чи не кидається в очі колір. Якщо представляють інтерес кольору укладені в деякому кубі (або гіперкубі при n> 3) з довжиною ребра WM з центром в точці (a1, а2, ..., аn) колірного простору. яка відповідає деякому заданому кольору -прототипу, то сукупність необхідних функцій перетворення задається виразом



Ці перетворення виділяють кольори навколо заданого, замінюючи всі інші на колір середньої точки використовуваного колірного простору. В випадку колірного простору RGB підходящою точкою є середина відрізка сірих кольорів, тобто колір (0,5,0,5,0,5).

Якщо для задання кольорової області, яка нам потрібна, використовується сфера, то верхня формула набуває наступного вигляду

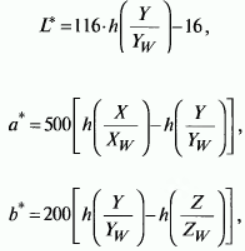


Тут R- радіус сфери (або гіперсфери при n>3), а (a1, a2, …, an) – координати сфери в колірному просторі, який визначає колір-прототип. В інших корисних модифікаціях формул використовуються кілька кольорів-прототипів, а інтенсивність кольорів, які нас не цікавлять, зменшуюються – замість того, щоб змінити ці кольори на нейтральні.

**1.3.2. Зміна яскравості**

Кольорові перетворення здійснюються на більшості персональних комп'ютерів. У поєднанні з цифровими фотокамерами, планшетними сканерами і кольоровими принтерами вони перетворюють персональний комп'ютер в цифрову фотолабораторію, яка дозволяє здійснювати вирівнювання яскравості і колірну корекцію (дві операції, що становлять основу будь-якої системи високоякісної кольорової репродукції) без застосування традиційних засобів хімічної обробки, використовуваних у звичайній фотолабораторії. Хоча зміна яскравості і колірна корекція корисні в різних областях обробки зображень, в центрі нашого обговорення буде перебувати їх найбільш поширене застосування - поліпшення фотографій і кольорове репродукування.

Ефективність перетворень відтінюється виключно за результатами друку. При розробці та вдосконаленні цих перетворень відтінку здійснюється з використанням монітора, тому необхідно забезпечити високу ступінь колірної відповідності між використовуваними моніторами і можливими пристроями виведення. Насправді, монітор повинен точно відтворювати кольори вихідного зображення, представленого в цифровому форматі, так само як і остаточні кольори зображення в тому вигляді, як вони з'являться на друку. Найкращим чином це досягається при використанні незалежної від колірної моделі. яка пов'язує між собою колірні охоплення моніторів, пристроїв виводу, а також інших використовуваних пристроїв. Успіх такого підходу визначається як якістю колірного профілю, використовуваного для відображення кожного з пристроїв в колірну модель, так і моделлю як такою. У ролі такої моделі в багатьох системах керування кольором виступає МКО (СE) модель L\*a\*b\*, так звана моделлю CIELAB. Координати кольорів в моделі L\*a\*b\* задаються наступними виразами:



де



a величини X, Y, і Z є координатами опорного білого кольору. В якості такого використовується біле світло, відбите ідеальною дифузною поверхнею, освітленою джерелом D65 стандарту МKO. Простір L\*a\*b\* є колориметричним (тобто, кольори, які сприймаються однаково, мають однакові координати). Хоча цей колірний простір не може бути відображено безпосередньо (для відображення на екрані або при друці необхідний перехід в інший колірний простір), його колірне охоплення включає весь видимий спектр і дозволяє точно уявити кольори будь-яких моніторів, принтерів і інших пристроїв введення-виведення. Подібно системі HSI, система L\*a\*b\* чудово розділяє інтенсивність (яка представлена ​​яскравістю L\*) і кольоровість (яка представлена ​​двома частинами: a\* - червоний мінус зелений і b\* - зелений мінус синій). Ця властивість робить систему L\*a\*b\* дуже зручною як для поліпшення зображень (тональної і колірної корекції), так і для їх стиснення.

Головна перевага каліброваних систем обробки зображень полягає в тому. що вони дозволяють здійснювати яскравісну і колірну корекції в інтерактивному режимі незалежно, тобто у вигляді двох послідовних операцій. Спочатку звичайно виробляються корекція яскравості діапазону зображення, а потім усувається колірний дисбаланс, такий як недостатня або надлишкова насиченість кольорів. Діапазон яскравості кольорового зображення пов'язаний із загальним розподілом інтенсивності його кольорів. Як і в монохромному випадку, часто бажано розподілити значення інтенсивності кольорового зображення рівномірно між місцями найбільш світлими і найбільш темними.

**ЗАВДАННЯ ДО РОБОТИ**

1.Запустити код для колірної обробки інформації у зображеннях.

2.Виокремити з програми один із сегментів що стосується окремих кольорів (зелений, синій, червоний), градацій сірого та яскравості або контрасту.

3.Здійснити обробку інформації за допомогою отриманої програми.

**ФРАГМЕНТ КОДУ**

namespace Image

{

partial class Form1

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

this.pictureBox1 = new System.Windows.Forms.PictureBox();

this.groupBox1 = new System.Windows.Forms.GroupBox();

this.contrastTrackBar = new System.Windows.Forms.TrackBar();

this.contrastLabel = new System.Windows.Forms.Label();

this.brightnessTrackBar = new System.Windows.Forms.TrackBar();

this.brightnessLabel = new System.Windows.Forms.Label();

this.menuStrip1 = new System.Windows.Forms.MenuStrip();

this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.openToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.saveToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.saveAsToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.openFileDialog1 = new System.Windows.Forms.OpenFileDialog();

this.saveFileDialog1 = new System.Windows.Forms.SaveFileDialog();

this.groupBox2 = new System.Windows.Forms.GroupBox();

this.button5 = new System.Windows.Forms.Button();

this.button4 = new System.Windows.Forms.Button();

this.label2 = new System.Windows.Forms.Label();

this.button3 = new System.Windows.Forms.Button();

this.label1 = new System.Windows.Forms.Label();

this.colorComponents = new System.Windows.Forms.CheckedListBox();

this.button2 = new System.Windows.Forms.Button();

this.button1 = new System.Windows.Forms.Button();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();

this.groupBox1.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.contrastTrackBar)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.brightnessTrackBar)).BeginInit();

this.menuStrip1.SuspendLayout();

this.groupBox2.SuspendLayout();

this.SuspendLayout();

//

// pictureBox1

//

this.pictureBox1.Anchor = ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Bottom)

| System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right)));

this.pictureBox1.BackColor = System.Drawing.SystemColors.ControlDark;

this.pictureBox1.Location = new System.Drawing.Point(0, 24);

this.pictureBox1.Name = "pictureBox1";

this.pictureBox1.Size = new System.Drawing.Size(499, 516);

this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;

this.pictureBox1.TabIndex = 0;

this.pictureBox1.TabStop = false;

//

// groupBox1

//

this.groupBox1.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));

this.groupBox1.BackColor = System.Drawing.SystemColors.Control;

this.groupBox1.Controls.Add(this.contrastTrackBar);

this.groupBox1.Controls.Add(this.contrastLabel);

this.groupBox1.Controls.Add(this.brightnessTrackBar);

this.groupBox1.Controls.Add(this.brightnessLabel);

this.groupBox1.Location = new System.Drawing.Point(505, 24);

this.groupBox1.Name = "groupBox1";

this.groupBox1.Size = new System.Drawing.Size(197, 164);

this.groupBox1.TabIndex = 1;

this.groupBox1.TabStop = false;

this.groupBox1.Text = "Properties";

this.groupBox1.Visible = false;

//

// contrastTrackBar

//

this.contrastTrackBar.Anchor = ((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right)));

this.contrastTrackBar.Location = new System.Drawing.Point(10, 107);

this.contrastTrackBar.Maximum = 255;

this.contrastTrackBar.Name = "contrastTrackBar";

this.contrastTrackBar.Size = new System.Drawing.Size(181, 42);

this.contrastTrackBar.TabIndex = 3;

this.contrastTrackBar.TickFrequency = 16;

this.contrastTrackBar.Value = 128;

this.contrastTrackBar.Scroll += new System.EventHandler(this.contrastTrackBar\_Scroll);

//

// contrastLabel

//

this.contrastLabel.AutoSize = true;

this.contrastLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.contrastLabel.Location = new System.Drawing.Point(6, 84);

this.contrastLabel.Name = "contrastLabel";

this.contrastLabel.Size = new System.Drawing.Size(83, 20);

this.contrastLabel.TabIndex = 2;

this.contrastLabel.Text = "Contrast 0";

//

// brightnessTrackBar

//

this.brightnessTrackBar.Anchor = ((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right)));

this.brightnessTrackBar.Location = new System.Drawing.Point(10, 39);

this.brightnessTrackBar.Maximum = 255;

this.brightnessTrackBar.Minimum = -255;

this.brightnessTrackBar.Name = "brightnessTrackBar";

this.brightnessTrackBar.Size = new System.Drawing.Size(181, 42);

this.brightnessTrackBar.TabIndex = 1;

this.brightnessTrackBar.TickFrequency = 32;

this.brightnessTrackBar.Scroll += new System.EventHandler(this.brightnessTrackBar\_Scroll);

//

// brightnessLabel

//

this.brightnessLabel.AutoSize = true;

this.brightnessLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.brightnessLabel.Location = new System.Drawing.Point(6, 16);

this.brightnessLabel.Name = "brightnessLabel";

this.brightnessLabel.Size = new System.Drawing.Size(98, 20);

this.brightnessLabel.TabIndex = 0;

this.brightnessLabel.Text = "Brightness 0";

//

// menuStrip1

//

this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.fileToolStripMenuItem});

this.menuStrip1.Location = new System.Drawing.Point(0, 0);

this.menuStrip1.Name = "menuStrip1";

this.menuStrip1.Size = new System.Drawing.Size(702, 24);

this.menuStrip1.TabIndex = 2;

this.menuStrip1.Text = "menuStrip1";

//

// fileToolStripMenuItem

//

this.fileToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.openToolStripMenuItem,

this.saveToolStripMenuItem,

this.saveAsToolStripMenuItem});

this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";

this.fileToolStripMenuItem.Size = new System.Drawing.Size(35, 20);

this.fileToolStripMenuItem.Text = "File";

//

// openToolStripMenuItem

//

this.openToolStripMenuItem.Name = "openToolStripMenuItem";

this.openToolStripMenuItem.ShortcutKeys = ((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Control | System.Windows.Forms.Keys.O)));

this.openToolStripMenuItem.Size = new System.Drawing.Size(180, 22);

this.openToolStripMenuItem.Text = "Open";

this.openToolStripMenuItem.Click += new System.EventHandler(this.openToolStripMenuItem\_Click);

//

// saveToolStripMenuItem

//

this.saveToolStripMenuItem.Name = "saveToolStripMenuItem";

this.saveToolStripMenuItem.ShortcutKeys = ((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Control | System.Windows.Forms.Keys.S)));

this.saveToolStripMenuItem.Size = new System.Drawing.Size(180, 22);

this.saveToolStripMenuItem.Text = "Save";

this.saveToolStripMenuItem.Click += new System.EventHandler(this.saveToolStripMenuItem\_Click);

//

// saveAsToolStripMenuItem

//

this.saveAsToolStripMenuItem.Name = "saveAsToolStripMenuItem";

this.saveAsToolStripMenuItem.ShortcutKeys = ((System.Windows.Forms.Keys)(((System.Windows.Forms.Keys.Control | System.Windows.Forms.Keys.Shift)

| System.Windows.Forms.Keys.S)));

this.saveAsToolStripMenuItem.Size = new System.Drawing.Size(180, 22);

this.saveAsToolStripMenuItem.Text = "Save as";

this.saveAsToolStripMenuItem.Click += new System.EventHandler(this.saveAsToolStripMenuItem\_Click);

//

// openFileDialog1

//

this.openFileDialog1.FileName = "openFileDialog1";

this.openFileDialog1.Filter = "PNG|\*.png|JPEG|\*.jpg|Bitmap|\*.bmp|GIF|\*.gif";

//

// saveFileDialog1

//

this.saveFileDialog1.Filter = "PNG|\*.png|JPEG|\*.jpg|Bitmap|\*.bmp|GIF|\*.gif";

//

// groupBox2

//

this.groupBox2.Anchor = ((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Bottom)

| System.Windows.Forms.AnchorStyles.Right)));

this.groupBox2.Controls.Add(this.button5);

this.groupBox2.Controls.Add(this.button4);

this.groupBox2.Controls.Add(this.label2);

this.groupBox2.Controls.Add(this.button3);

this.groupBox2.Controls.Add(this.label1);

this.groupBox2.Controls.Add(this.colorComponents);

this.groupBox2.Controls.Add(this.button2);

this.groupBox2.Controls.Add(this.button1);

this.groupBox2.Location = new System.Drawing.Point(505, 194);

this.groupBox2.Name = "groupBox2";

this.groupBox2.Size = new System.Drawing.Size(197, 346);

this.groupBox2.TabIndex = 3;

this.groupBox2.TabStop = false;

this.groupBox2.Text = "Effects";

this.groupBox2.Visible = false;

//

// button5

//

this.button5.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.button5.Location = new System.Drawing.Point(55, 242);

this.button5.Name = "button5";

this.button5.Size = new System.Drawing.Size(136, 28);

this.button5.TabIndex = 8;

this.button5.Text = "Flip bottom to top";

this.button5.UseVisualStyleBackColor = true;

this.button5.Click += new System.EventHandler(this.button5\_Click);

//

// button4

//

this.button4.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.button4.Location = new System.Drawing.Point(55, 208);

this.button4.Name = "button4";

this.button4.Size = new System.Drawing.Size(136, 28);

this.button4.TabIndex = 7;

this.button4.Text = "Flip left to right";

this.button4.UseVisualStyleBackColor = true;

this.button4.Click += new System.EventHandler(this.button4\_Click);

//

// label2

//

this.label2.AutoSize = true;

this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label2.Location = new System.Drawing.Point(6, 185);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(81, 20);

this.label2.TabIndex = 6;

this.label2.Text = "Flip image";

//

// button3

//

this.button3.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.button3.Location = new System.Drawing.Point(79, 118);

this.button3.Name = "button3";

this.button3.Size = new System.Drawing.Size(112, 49);

this.button3.TabIndex = 4;

this.button3.Text = "Apply";

this.button3.UseVisualStyleBackColor = true;

this.button3.Click += new System.EventHandler(this.button3\_Click);

//

// label1

//

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label1.Location = new System.Drawing.Point(6, 95);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(182, 20);

this.label1.TabIndex = 3;

this.label1.Text = "Active color components";

//

// colorComponents

//

this.colorComponents.FormattingEnabled = true;

this.colorComponents.Items.AddRange(new object[] {

"Red",

"Green",

"Blue"});

this.colorComponents.Location = new System.Drawing.Point(10, 118);

this.colorComponents.Name = "colorComponents";

this.colorComponents.RightToLeft = System.Windows.Forms.RightToLeft.No;

this.colorComponents.Size = new System.Drawing.Size(63, 49);

this.colorComponents.TabIndex = 2;

//

// button2

//

this.button2.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.button2.Location = new System.Drawing.Point(10, 53);

this.button2.Name = "button2";

this.button2.Size = new System.Drawing.Size(181, 28);

this.button2.TabIndex = 1;

this.button2.Text = "Convert to grayscale";

this.button2.UseVisualStyleBackColor = true;

this.button2.Click += new System.EventHandler(this.button2\_Click);

//

// button1

//

this.button1.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.button1.Location = new System.Drawing.Point(10, 19);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(181, 28);

this.button1.TabIndex = 0;

this.button1.Text = "Make image negative";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.SystemColors.Control;

this.ClientSize = new System.Drawing.Size(702, 540);

this.Controls.Add(this.groupBox2);

this.Controls.Add(this.groupBox1);

this.Controls.Add(this.pictureBox1);

this.Controls.Add(this.menuStrip1);

this.MainMenuStrip = this.menuStrip1;

this.Name = "Form1";

this.Text = "Picture Editor";

this.Load += new System.EventHandler(this.Form1\_Load);

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();

this.groupBox1.ResumeLayout(false);

this.groupBox1.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.contrastTrackBar)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.brightnessTrackBar)).EndInit();

this.menuStrip1.ResumeLayout(false);

this.menuStrip1.PerformLayout();

this.groupBox2.ResumeLayout(false);

this.groupBox2.PerformLayout();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.PictureBox pictureBox1;

private System.Windows.Forms.GroupBox groupBox1;

private System.Windows.Forms.MenuStrip menuStrip1;

private System.Windows.Forms.ToolStripMenuItem fileToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem openToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem saveToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem saveAsToolStripMenuItem;

private System.Windows.Forms.OpenFileDialog openFileDialog1;

private System.Windows.Forms.SaveFileDialog saveFileDialog1;

private System.Windows.Forms.Label brightnessLabel;

private System.Windows.Forms.TrackBar brightnessTrackBar;

private System.Windows.Forms.TrackBar contrastTrackBar;

private System.Windows.Forms.Label contrastLabel;

private System.Windows.Forms.GroupBox groupBox2;

private System.Windows.Forms.Button button2;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.CheckedListBox colorComponents;

private System.Windows.Forms.Button button3;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Button button5;

private System.Windows.Forms.Button button4;

}

}