

YUV (Raw Data) ÇÖZÜCÜ

Kocaeli Üniversitesi

Mühendislik Fakültesi

Bilgisayar Mühendisliği

Muhammed Eren Uygun – Mehmet Akif Altınok

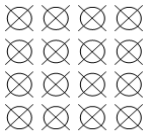
erenn.uygun@gmail.com – akif.altinok2@gmail.com

160201064 – 160201029

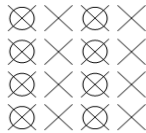
YUV nedir? Yuv renk formatı RGB formatına alternatif olarak sunulan bir renk formatıdır. **YUV**, genellikle video görüntülerini kaydetmek için kullanılan bir renk sistemidir.

Y: *Luminance*, **U**: *Chrominance1*, **V**: *Chrominance2* sözcüklerinin baş harflerinden oluşan kısaltmadır. Yani YUV formatının **Y** bileşeni Luma yani, ışıklılık bileşenine denk düşer. **U** bileşeni Chroma yani renklilik bileşenine karşılık gelir. **V** bileşeni de Chroma yani renklilik bileşenine karşılık gelir. **Y** ve **U-V** bileşenleri, ışıklılık ve renk kanalları olarak da bilinmektedir. Ayrıca RGB renk uzayı, İngilizcedeki 'Red' 'Green' 'Blue', yani 'Kırmızı' 'Yeşil' 'Mavi' kelimelerinin baş harflerinden ismini alan bir renk uzayı olup en sık kullanılanlardandır. Işığın temel olarak, doğadaki tüm renklerin kodları bu üç temel renge referansla belirtilir. **RGB** renk uzayında bir piksel ortalama 24 bit ile ifade edilirken, **YUV** renk uzayında piksel başına ortalama 24 bitten daha az sayıda bit kullanılabilmektedir. Bunun nedeni, **U** ve **V** kanallarında **Y** kanalına göre daha az sayıda piksel içermesidir. Literatürde farklı örnekleme türleriyle hazırlanmış YUV formatları bulunmaktadır. Biz sadece bu formatların 3 tanesi ile ilgileneceğiz. Bunlar ise 4-4-4 formatı, 4-2-2 formatı ayrıca 4-2-0 formatları olacaktır.

- **4:4:4** formatı: Renk kanallarında (Chroma kanallarında, **Cr-Cb**) herhangi bir alt örnekleme yapılmamıştır.
- **4:2:2** formatı: Renk kanallarında, $\frac{1}{2}$ oranında yatay alt-örnekleme yapıldığı, düşeyde ise herhangi bir alt-örnekleme yapılmadığı duruma karşılık gelmektedir.
- **4:2:0** formatı: Renk kanallarında, $\frac{1}{2}$ oranında yatay alt-örnekleme ve düşey alt-örnekleme yapıldığı durumudur.



4:4:4

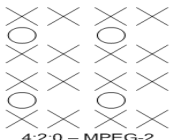


4:2:2 – BT.601

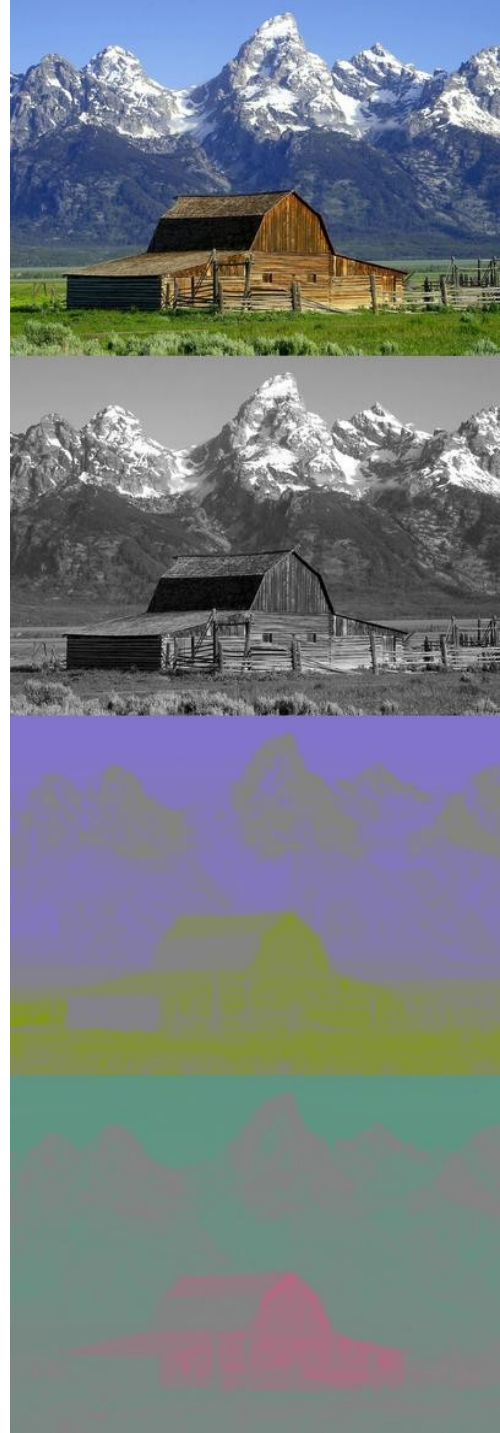


Luma

Chroma



4:2:0 – MPEG-2



I. GİRİŞ

Projemizi .yuv formatının çözümü ile ilgilidir. Bu projemizde elimizde bulunan .yuv uzantılı dosyaların okunması ve daha sonrasında bu okuduğumuz içeriğin byte'a çevirilmesi gerçekleşiyor. Byte çevirdiğimiz içeriği bir döngü içerisinde frame frame Y,U ve V bileşenlerini alıyoruz. Bu bileşenlerimizi de RGB formatına dönüştürüp sadece Y bileşeni olacak şekilde yani videomuzun sadece siyah-beyaz görüneceği şekilde düzenliyoruz ve frame frame formumuzu bastırıyoruz. Arada da delay olduğu için videomuz tamamlanmış oluyor. Biz bu projemizi kullanırken bizden istenilen programlama dilleri arasında c# programlama dilini tercih ettik.

C#, yazılım sektörü içerisinde en sık kullanılan iki yazılım dili olan C ve C++ etkileşimi ile türetilmiştir. Ayrıca C#, ortak platformlarda taşınabilir bir (portable language) programlama dili olan Java ile pek çok açıdan benzerlik taşımaktadır. En büyük özelliği ise .Net Framework platformu için hazırlanmış tamamen nesne yönelimli bir yazılım dilidir. Yani nesneler önceden sınıflar halinde yazılıdır. Programcıya sadece o nesneyi sürüklemek ve sonrasında nesneyi amaca uygun çalıştıracak kod satırlarını yazmak kalır.

Microsoft tarafından geliştirilen C#, C++ ve Visual Basic dillerinde yer alan tutarsızlıkları kaldırmak için geliştirilmiş bir dil olmasına rağmen kısa süre içerisinde nesne yönelimli dillerin içinde en gelişmiş programlama dillerinden biri olmayı başarmıştır. Ayrıca gelişmiş derleyicisi (debugger) ile hata olasılığını ortadan kaldırmaktadır. Yazılan program çalıştırıldıktan sonra derleyici tarafından algılanan Sınıf (Class) ve söz dizimi (syntax) hataları yazılımcıya ayrı bir ekranda ayrıntısı ile gösterilir ve yazılımcı bu hata penceresinden hataları tespit ederek kolayca düzeltebilir.

Ayrıca pek çok programcı tarafından kullanılan bir dil olmasından ötürü takıldığımız noktalarda uzman programcılardan yardım almak oldukça kolaydır.

C# Ne işe Yarar?

Yeni nesil programlama dili olan C sharp programlama konusunda yenilikleri de beraberinde getirmektedir. C# sunucu ve gömülü sistemleri için tasarlanmıştır. Microsoft tarafından geliştirilen C# .NET ortamında kullanılmak üzere geliştirilmiştir. Dilin tasarlanmasında Anders Hejlsberg öncülük etmiştir. Zamanla gelişen bu programlama dili artık gerek masaüstü olsun gerek web uygulamaları olsun çoğu yerde kullanılmaktadır.

.NET Framework Nedir?

C# ve .Net Framework bazı kişiler tarafından tek bir kavram olarak algılanmaktadır. Fakat bu iki kavram birbirlerinden tamamen farklı amaçlar için geliştirilmiştir. C#, nesne yönelimli bir programlama diliyken .Net Framework ise C# için geliştirilmiş bir çalışma ortamıdır. Aslında C# dili, Microsoft tarafından .Net platformu için kod geliştirmek

amaçlı tasarlanmış ve C# içerisindeki tüm kütüphaneler .Net platformu içinde tanımlanmış kütüphanelerdir.

Java'dan önce, geliştirilen yazılımlar makine koduna çevrilerek çalıştırılırdı. Java ise program kodlarını önce byte sayı sistemine çevirir. Sonrasında Java Sanal Makinesi (JVM —Java Virtual Machine) bu kodları işletim sisteminin istediği koda çevirerek programın çalışmasına sağlar.

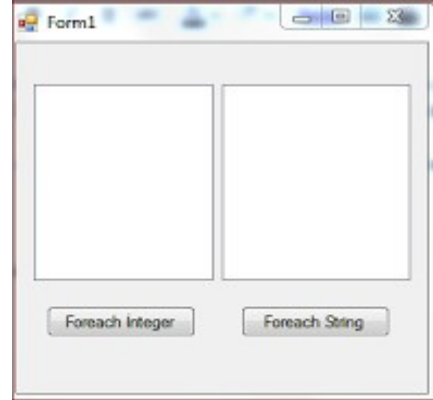
.Net Platformu da Java diline benzer bir çalışma mantığı izleyerek kodları çalışabilir hale getirmektedir.

.Net platformunda kod ilk önce Microsoft Intermediate Language (Microsoft Ara Dili) olarak isimlendirilmiş dosya haline dönüştürülür bu dosya içerisinde derlenen kodların Microsoft'un standart haline getirdiği bir assembly dili haline dönüştürür. Bu ara dil de saklanan dosyalar çalıştırılmak istendiğinde ise CLR adı verilen sistem MSIL kodlarını çalıştırır.

Biz projemiz windows form uygulamalarını kullandık.

Windows Form Uygulamaları (Ado.Net)

Grafiksel kullanıcı arayüzü (GUI—Graphical User Interface) olarak tanımlanır. Windows form uygulamalarını, konsol uygulamalarından ayıran en önemli özelliği de budur. Windows Form uygulaması oluşturmak için Toolbox adı verilen araç kutusundan nesnel araçlar sürükleyip bırak yöntemi ile kolayca oluşturulabilir.



Windows Form uygulamaları, bilgisayar programları geliştirmek amacıyla kullanılmaktadır. Windows Forms Application seçeneği ile yeni bir windows uygulama penceresi açılarak program geliştirmeye başlanabilir. Ayrıca Windows Form uygulaması içinde grafiksel olarak daha gelişmiş çalışmalar yapmak da mümkün hale getirilmiştir. Daha güçlü grafiksel arayüzler için WPF ile uygulama geliştirmek gerekmektedir.

```
Program.cs
1 using System;
2
3 namespace DotNetBot {
4
5     public static class Program {
6
7         public static void Main(string[] args) {
8
9             string message = "";
10             if (args.Length < 1) {
11                 message = "Welcome to .NET Core!";
12             }
13             else {
14                 foreach (string item in args) {
15                     message += item;
16                 }
17             }
18         }
19     }
20 }
```

II. .YUV FORMATLI DOSYANIN OKUNMASI

.yuv uzantılı dosyamızı projemizi açtıktan sonra dosya seç butonuna basarak seçiyoruz. Bu butondan çektiğimiz .yuv uzantılı dosyanın yolunu bir FileStream değişkenine atıyoruz. Bu arada FileStream nesnesini kullanabilmek için system.io kütüphanesini sistemimize dahil ediyoruz. Daha sonra dosyamızın içeriğini frame frame okuyabilmemiz için bir while döngüsü içerisinde filestream'i kullanarak her bir döngüde bir frame okuyoruz. Bir video dosyası aslında karelerden oluşmaktadır. Ve bu kareler bir araya gelerek ise video oluşur. Frame ise bu videolardaki karelere verilen isimdir. Bizim amacımız ise kullanıcıdan almış olduğumuz widthXheight(frame) yani genişlikXyükseklik değerini okumaktır. Eğer .yuv uzantılı dosyamızın formatı 4-4-4 ise Y,U ve V bileşenleri sırasıyla widthXheight/3, widthXheight/3, widthXheight/3 'ü kadardır. Eğer .yuv uzantılı dosyamızın formatı 4-2-2 ise Y,U ve V bileşenleri sırasıyla widthXheight/3, widthXheight/6, widthXheight/6 'sı kadardır. Eğer .yuv uzantılı dosyamızın formatı 4-2-0 ise Y,U ve V bileşenleri sırasıyla widthXheight/3, widthXheight/12, widthXheight/12 'si kadardır. Her bir frame'i okuduktan sonra bu frame'i byte formatına dönüştürüyoruz. Byte formatına çevirirken C#'ın bize sunmuş olduğu nesnelerden olan Bitmap nesnesini kullanıyoruz. Bu nesneyi kullanabilmemiz için system.drawing kütüphanesini dahil etmemiz gerekmektedir. Ardından frame'i Y,U ve V bileşenlerine ayırma işlemini yapıyoruz. Bu projede bizden istenen sadece Y bileşenleridir. Yani göstereceğimiz framerler sadece Y bileşeni olduğu için siyah-beyaz görünecektir. O halde parçalamış olduğumuz ve byte dönüştürdüğümüz bileşenlerinden sadece Y bileşenini YUVtoRGB fonksiyonuna gönderiyoruz. Böylece .yuv uzantılı dosyamızın Y,U ve V bileşenleri arasından sadece Y bileşenini okuyup byte a çevirip onu da fonksiyon yordamıyla RGB formatına dönüştürüp kullanıma hazır hale getirmiş olduk.

```
using (FileStream fs = File.OpenRead(textBox1.Text))
{
    int frame = (int)fs.Length / frameSize;
    using (BinaryReader br = new BinaryReader(fs))
    {
        while (br.PeekChar() != -1)
        {
            br.Read(yuv, 0, frameSize);
            bm = YUVtoRGB(yuv, rgb, width, height);
            l.Add(bm);
            imageList1.Images.Add(bm);
            index++;
        }
    }
    System.Windows.Forms.MessageBox.Show("Okuma işlemi başarılı!");
    oynat.Enabled = true;
    durdur.Enabled = true;
    bmp_kaydet.Enabled = true;
}
```

III. YUVtoRGB FONKSİYONU

Dosyadan okumuş olduğumuz Y,U ve V bileşenlerini YUVtoRGB fonksiyonuna gönderiyoruz. YUV formatını RGB formatına dönüştürebilmemiz için Y,U ve V değerlerini statik sayılar ile çarpmamız gerekir. Bu projede sadece Y bileşenlerine ihtiyacımız olduğu için R,G ve B formatlarının hepsine Y bileşenini gönderiyoruz. Bu Y bileşenlerini her zaman aynı olan katsayımız ile çarpıyoruz. Daha sonra çarpmış olduğumuz değeri bir değişkene atıyoruz. Ardından Color.FromArgb fonksiyonunu kullanarak sırası ile elde ettiğimiz yeni R,G ve B bileşenlerini fonksiyona gönderiyoruz. Dönen değeri ise Bitmap formatında oluşturduğumuz nesnenin setPixel kısmına gönderiyoruz. Ardından piksellerini oluşturduğumuz Bitmap nesnesini main fonksiyonuna gönderiyoruz. Main fonksiyonunda ise Bitmap olarak gönderdiğimiz değeri bir Bitmap listesine atıyoruz. Böylelikle bütün Y bileşenlerini bir Bitmap listesinde tutabiliyoruz.

```
static Bitmap YUVtoRGB(byte[] y, byte[] rgbFrame, int width, int height)
{
    int gIndex = width * height;
    int bIndex = gIndex * 2;
    int temp = 0;
    Bitmap bm = new Bitmap(width, height);
    for (int i = 0; i < height; i++)
    {
        for (int j = 0; j < width; j++)
        {
            // R
            temp = (int)((y[i * width + j]) * 1.4822);
            rgbFrame[i * width + j] = (byte)(temp < 0 ? 0 : (temp > 255 ? 255 : temp));
            // G
            temp = (int)((y[i * width + j]) * -0.3456);
            rgbFrame[gIndex + i * width + j] = (byte)(temp < 0 ? 0 : (temp > 255 ? 255 : temp));
            // B
            temp = (int)((y[i * width + j]) * 1.771);
            rgbFrame[bIndex + i * width + j] = (byte)(temp < 0 ? 0 : (temp > 255 ? 255 : temp));
            //Pixel yazdır
            Color c = Color.FromArgb(rgbFrame[i * width + j], rgbFrame[i * width + j], rgbFrame[i * width + j]);
            bm.SetPixel(j, i, c);
        }
    }
    return bm;
}
```

KAYNAKLAR

- [1] stackoverflow.com
- [2] <http://www.wikizero.net/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvWVVW>
- [3] <https://docs.microsoft.com/tr-tr/dotnet/>
- [4] <https://stackoverflow.com/questions/5194285/yuv-file-format>
- [5] <https://code.visualstudio.com/docs/languages/csharp>
- [6] <https://stackoverflow.com/questions/23041834/convert-yuv-file-to-image-in-c-sharp>
- [7] <https://msdn.microsoft.com/en-us/library/aa917087.aspx>
- [8] [https://docs.microsoft.com/en-us/previous-versions/windows/embedded/ee490095\(v=winembedded.60\)](https://docs.microsoft.com/en-us/previous-versions/windows/embedded/ee490095(v=winembedded.60))