

YUV ÇÖZÜCÜ

Ata GÜLALAN - Oğuzhan TÜRKER

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

xavaneo@gmail.com oguzturker8@gmail.com

Özet

Uygulama, kullanıcı tarafından seçilen bir YUV formatlı dosyayı okuyarak RGB renk uzayına çevirir. RGB renk uzayındaki bir veya birden fazla çerçeve, kullanıcı tarafından görüntülenebilir, oynatılabilir, kaydedilebilir.

1. Giriş

Uygulama ilk açıldığında, kullanıcıdan bir YUV formatında bir dosya açmasını bekler. Arayüzde bulunan “File” sekmesinde bulunan “Open” seçeneğinden dosya yükleme ekranı açılır. Seçilen YUV formatlı dosya programa yüklenir.

Dosya açıldığında, öntanımlı olarak CIF (352x288) boyutlarında, YUV 4:2:0 formatı seçili olarak okunur. Okunan dosyanın boyutu, format ve boyutlara bağlı olarak hesaplanan boyuttan küçük ise dosya okunmaz. Dosya okunduktan sonra maksimum çerçeve sayısı hesaplanır. YUV dosyası okunduktan sonra bir değişkene atandığından, ilk çerçeveden sonraki çerçevelerin hesaplanması için tekrar dosyadan okunmasına gerek yoktur. Maksimum çerçeve sayısı, okunan dosyanın boyutunun, istenen format ve boyuta bağlı olan bir değişkene bölünmesi ile bulunur.

Uygulama her bir çerçeve için; verilen format ve boyutu, istenen çerçeveye kadar kaydırarak o çerçevenin verilerine erişir. Görüntü her bir çerçeve için RGB formatına dönüştürülür. Dönüştürülen görüntü ekranda gösterilir. Yüklenen dosyanın formatı veya boyutu, öntanımlı verilerden farklı ise arayüzde bulunan “Size” ve “Color” menüleri ile ilgili dosyanın formatı ve boyutu girilebilir.

2. Temel Bilgiler

Proje gelişiminde; Programlama dili olarak “C#”, tümleşik geliştirme ortamı olarak “Visual Studio 2017” kullanılmıştır. Uygulama içeriği olarak “Windows Presentation Foundation (WPF)” kullanılmıştır.

3. Tasarım

Proje aşağıdaki başlıklar altında geliştirilmiştir.

3.1 Yazılım Tasarımı

Projenin yazılım aşaması bu başlık altında bulunan konular tarafınca geliştirilmiştir.

3.1.1 Genel Değişkenler

- **player** Aktif iken, bir milisaniye aralıklarla bir sonraki çerçeveyi sahneler.
- **maxFrame** Açılan dosyanın, istenen format ve boyutun kaç katına denk geldiğini tutar.
- **width** Çerçeve genişliğini tutar.
- **height** Çerçeve yüksekliğini tutar.
- **type** Çerçeve tipini tutar.
- **bytearr** Okunan YUV dosyasını, byte dizisi şeklinde tutar.
- **bitmap** O anki çerçevenin Bitmap’e çevirilmiş halini tutar.

3.1.2 Fonksiyonlar

Uygulamada bulunan fonksiyonlar aşağıda listelenmiştir.

3.1.2.1 İşlem Fonksiyonları

- **ArraySplitter:** Gelen byte dizisini istenilen şekilde bölüp ayırır.
- **Limiter:** Alınan tam sayıyı 0’dan küçük ve 255’den büyük olmayacak şekilde döndürür.
- **YUV2RGB:** Y, U ve V şeklinde gelen byte dizilerini alır ve ABGR (Alpha, Blue, Green, Red) formatına, buradan da bitmap’e çevirir.
- **RenderFrame:** Okunan byte dizisini Y, U ve V dizilerine ayırır. Ayrımı **YUV2RGB** fonksiyonuna gönderir. Geri dönen bitmap’i ekrana yazdırır.
- **SetSize:** Parametrelerdeki genişlik ve yüksekliği genel değişkenlere atar ve **RenderFrame** fonksiyonunu çağırır.

- **UnChecker:** Menülerde başka seçim yapıldığında önceki seçimi kaldırır ve tıklanan öğeyi seçer.
- **OpenFile:** Dosya okuma fonksiyonudur. Dosyayı okur ve bytearray genel değişkenine atar. İlk atamadan sonra ilk çerçeve sahnelenir. Size ve Color menüleri bu işlem sonrası aktifleşir.

3.1.2.2 Arayüz Fonksiyonları

- **Open_Click:** YUV formatındaki dosyayı okuma penceresini açar.
- **Save_Click** Çerçeveyi kaydetmek için kayıt penceresi açar. Eğer herhangi bir çerçeve görüntülenmiyorsa pencere açılmaz.
- **Exit_Click** Tüm uygulamadan çıkış yapar.
- **OpenSizeDialog_Click** Kullanıcı tarafından belirlenecek görüntü boyut giriş penceresini açar.
- **ChangeColor_Click:** YUV formatını kullanıcıdan alır ve alınan formata göre çerçeve tekrar sahnelenir.
- **PlayStop_Click:** Videoyu oynatır/durdurur.
- **Slider_ValueChanged** Görüntünün istenilen çerçevesini sahneler.

3.1.3 Çevrim

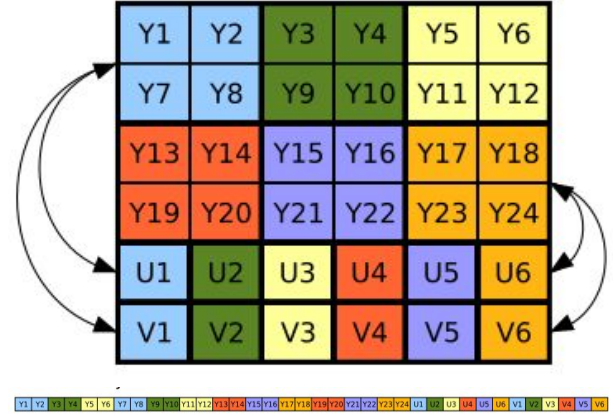
Programda toplam 3 adet çevrim kullanıldı. Bu çevrimler, YUV 4:4:4, YUV 4:2:2 ve YUV 4:2:0 için yapıldı.

Programda, her bir karenin kapladığı toplam boyut bulunması için her tipe bağlı olan etki değişkeni kullandık. Bu etki değişkeni, 4:4:4 için 3, 4:2:2 için 2, 4:2:0 için 1.5 idi. Bu değerler, Tüm blok boyutunun Y'ye oranlanması ile bulundu.

Kareleri ayırdıktan sonra ayrıştırma işlemi için yine tipe bağlı hareket ettik:

- YUV 4:4:4 için U ve V, her bir Y değerine karşılık geldiğinden her bir Y değeri için U ve V değerlerini birebir aldık.
- YUV 4:2:2 için her iki Y değerine karşılık sadece bir U ve V olduğundan, her U ve V değerini iki Y değeri ile eşleştirdik. Bu eşleşme, yatayda bir kalite bozulmasına yol açmasa da, dikeyde oluşturulan chroma kanallarının yarım çözünürlükte olmasına sebep oldu. Bu formatı kullanarak yaklaşık %33 tasarruf ettik.
- YUV 4:2:0 için her dört Y değerine karşılık sadece bir U ve V olduğundan, her U ve V değerini dört Y değeri ile eşleştirdik. Bu eşleşme; hem dikeyde hem de yatayda, chroma kanallarının çeyrek çözünürlükte olmasına sebep oldu. Bu formatı kullanarak %50 tasarruf ettik.

Aşağıda, YUV 4:2:0 için çevrilen ışık ve chroma kanalları gözükmemektedir.



Şekil 1: YUV 4:2:0 için uyumlu blokların gösterimi

Hangi değerlerin nerede olduğunu ve nasıl sıkıştırıldığını öğrendiğimize göre YUV'dan RGB uzayına çevrimde sıkıntı çıkmayacak.

Aynı şekilde; kullanıcıdan alınan genişlik, yükseklik ve formata bağlı olarak bulunan değere bağlı olarak kaydırılarak birden fazla çerçevenin görünmesi sağlanıyor.

3.1.4 Arayüz

Program arayüzünde üç bölüm bulunmaktadır. Bölümler aşağıda numaralandırılmış halde sıralanmaktadır.

3.1.4.1 File (Dosya)

- **Open:** Okunacak dosya için okuma penceresi açar.
- **Save** Ekrandaki çerçeveyi kaydetmek için kayıt penceresi açar.
- **Exit** Programdan çıkış yapar.

3.1.4.2 Size (Boyut)

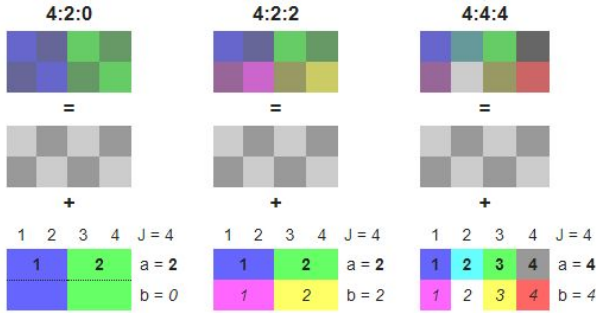
Uygulamada kullanılabilen görüntü boyutları aşağıda listelenmiştir.

- **1080p** 1920 x 1080 piksel görüntü boyutudur.
- **720p** 1280 x 720 piksel görüntü boyutudur.
- **576p** 720 x 576 piksel görüntü boyutudur.
- **VGA** 640 x 480 piksel görüntü boyutudur.
- **WVGA** 832 x 480 piksel görüntü boyutudur.
- **WQVGA** 352 x 288 piksel görüntü boyutudur.
- **CIF** 176 x 144 piksel görüntü boyutudur.
- **QCIF** 192 x 256 piksel görüntü boyutudur.
- **SMALL** 920 x 1080 piksel görüntü boyutudur.
- **Custom** Kullanıcı tarafından belirlenen görüntü boyutu seçeneğidir.

3.1.4.3 Color (Renk)

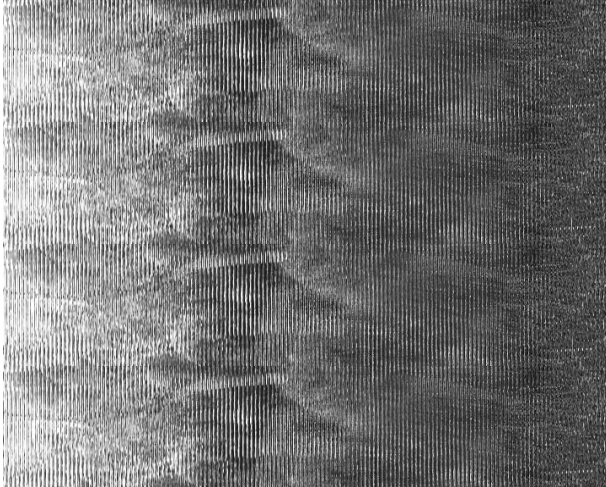
Uygulamada kullanılabilen YUV formatları aşağıda listelenmiştir.

- **YUV 4:4:4** Lumina (Işık) 4 Birim, Chroma1 (Renk 1) 4 Birim, Chroma2 (Renk 2) 4 Birim.
- **YUV 4:2:2** Lumina (Işık) 4 Birim, Chroma1 (Renk 1) 2 Birim, Chroma2 (Renk 2) 2 Birim.
- **YUV 4:2:0** Lumina (Işık) 4 Birim, Chroma1 (Renk 1) 1 Birim, Chroma2 (Renk 2) 1 Birim.



Şekil 2: YUV Formatları ve sıkıştırılma şekilleri

3.1.5 Karşılaşılan Sorunlar ve Çözüm Yöntemleri



Şekil 3: Genişlik ve Yüksekliğin Yanlış Taranması

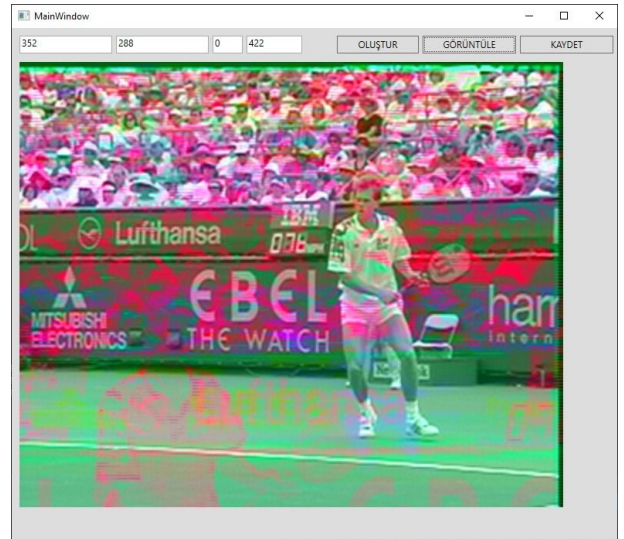
İlk hatalarımızdan biri, dosyanın yanlış taraftan taranmaya başlaması oldu. Her bir satırı soldan sağa taramak yerine, yukarıdan aşağıya taradık ve ortaya sanatsal bir görüntü çıktı (Şekil 3).

Bu sorunu ilk gördüğümüzde çok bir şey anlayamasa da, çizgilerin düzeni bize doğru yolda olduğumuzu gösterdi. Dosyayı HEX Editör ile açtığımızda gördüğümüz değerlerin soldan sağa değil de yukarıdan aşağıya doğru aktığını gördüğümüzde sorunun ne olduğunu anladık ve sorunu bu şekilde çözdük.



Şekil 4: Oversaturation (Aşırı Doyum) Problemi

Çevirim işleminden sonra kırmızı, yeşil ve mavi kanallardaki değerlerin bazılarının 0'dan az veya 255'den fazla olabileceğini keşfettik. Bu değerler, resim BitMap'e çevrildiğinde bazı pixellerin pembe olarak gözükmesine sebep oldu (Şekil 4). Bu sorunu, **Limiter** fonksiyonunu yazarak ve çevrilmiş her bir değeri bu fonksiyona yollayarak çözdük.



Şekil 5: 420 Kayma Problemi

Uygulamayı öncelikle YUV 4:4:4 okumak için geliştirdiğimizden genişlik ve yüksekliği ayrı ayrı tutmak yerine, döngüyü çarpımına kadar götürdük. Bu yazılan kodda YUV 4:2:2 formatını çalıştırmak için tek yapmamız gereken iki Y değeri için tek bir U ve V değerine bakmaktı. Ancak bu kodu YUV 4:2:0 için çalıştırmaya çalıştığımızda her iki satır ve sütunun tek bir U ve V değerine bağlı olmasından dolayı okuma işleminin tek bir boyutta yapamadık. Bu sorunu, her iki boyut için ayrı döngüler açıp U ve V'deki değerin hangi chroma kanallarının geleceğini belirleyerek çözdük.



Şekil 6: U ve V Kanallarının Karıştırılması

Sorunu görür görmez anladık. Mavi olması gereken yerler turuncu, turuncu/ten rengi olması gereken yerler ise mavi gözüküyordu (Şekil 6). Şirinler problemi olarak adlandırdığımız bu sorunu, YUV'dan RGB'ye dönüşüm sırasında kullandığımız U ve V kanallarının yanlış yere konulmasından olduğunu anladık ve yerini değiştirerek sorunu çözdük.

4. Kazanımlar

Kazanımlar aşağıda numaralandırılmıştır.

4.1 YUV Dosya Formatı

Yuv dosyası, yuv dosyasının okunup işlenmesi, yuv dosyasının formatları ve bu formatların kendine ait özellikleri.

4.2 RGB Dosya Formatı

Rgb dosya formatı, rgb dosyasının özellikleri, işleme ve saklanma yöntemleri.

4.3 Görüntü Formatları

1080p, 720p, 576p, VGA, WVGA, WQVGA, CIF, QCIF ve SMALL formatları.

4.4 Bitmap

Bitmap yapısı ve bitmap sınıfları.

5. Akış Şeması

bu sende
kanka



Şekil 7

6. Kaynakça

- [1] **Creating A Byte Array From A Stream**
<https://stackoverflow.com/questions/221925/creating-a-byte-array-from-a-stream>
Erişim Tarihi: 13.12.2018
- [2] **Convert Byte Array To Image In WPF**
<https://stackoverflow.com/questions/9564174/convert-byte-array-to-image-in-wpf>
Erişim Tarihi: 13.12.2018
- [3] **Array Copy in C#**
<https://www.tutorialspoint.com/Array-Copy-in-Csharp>
Erişim Tarihi: 13.12.2018
- [4] **C# How To Add Byte To Byte Array**
<https://stackoverflow.com/questions/5591329/c-sharp-how-to-add-byte-to-byte-array>
Erişim Tarihi: 13.12.2018
- [5] **Array.CopyTo Method (System)**
https://docs.microsoft.com/en-us/dotnet/api/system.array.copyto?redirectedfrom=MSDN&view=netframework-4.7.2#System_Array_CopyTo_System_Array_System_Int32_
Erişim Tarihi: 13.12.2018
- [6] **YUV to RGB Conversion**
<https://www.fourcc.org/fccyvrgb.php>
Erişim Tarihi: 13.12.2018
- [7] **Convert Bitmapsource To Bitmap**
<https://stackoverflow.com/questions/2284353/is-there-a-good-way-to-convert-between-bitmapsource-and-bitmap>
Erişim Tarihi: 13.12.2018
- [8] **Chroma Subsampling: 4:4:4, 4:2:2 and 4:2:0**
<https://www.rtings.com/tv/learn/chroma-subsampling>
Erişim Tarihi: 13.12.2018