

YUV (Raw Data) ÇÖZÜCÜ
Emre Altay, Batuhan Subaşı
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi

emrealltay@gmail.com, batuhansubasi@hotmail.com

Özet

YUV renk formatı, RGB renk uzayına alternatif olarak önerilmiş bir renk uzayıdır. Y bileşeni ışıklılık bileşenine, U (Cb) ve V (Cr) ise renk bileşenlerine karşılık gelmektedir. Y ve U-V bileşenleri, ışıklılık ve renk kanalları olarak da bilinmektedir. 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. Örneğin;

□ 4:4:4 formatı: Renk kanallarında (Chroma kanallarında, Cr-Cb) herhangi bir alt örnekleme yapılmamıştır.

□ 4:2:2 formatı: Renk kanallarında, 1/2 oranında yatay alt-örnekleme yapıldığı, düşeyde ise herhangi bir alt-örnekleme yapılmadığı duruma karşılık gelmektedir.

□ 4:2:0 formatı: Renk kanallarında, 1/2 oranında yatay alt-örnekleme ve düşey alt-örnekleme yapıldığı durumudur.

RGB'den YUV'a dönüşüm işleminden sonra .yuv uzantılı dosyalarda N adet çerçeve bilgisi bulunmaktadır. Her bir çerçeve için sırasıyla Y, U ve V bileşenleri tutulmaktadır. Şekil-1'de, ışıklılık (luma) ve renk bileşenleri (chroma) için hangi örneklerin/piksellerin seçilip/seçilmediği bilgisi gösterilmektedir.

Bizden istenenler bu projede aşağıdaki gibidir:

□ Arayüz tasarımı: Tasarlanacak arayüz üzerinden dosya okuma, .bmp kaydetme, görüntü oynatma ve ayarlar seçeneği olacaktır.

□ Dosya okuma: .yuv uzantılı dosyalar okunacak ve render/parse işlemine tabii tutulacaktır.

□ Ayarlar: Hangi formatta okuma işlemi yapılacağı ve görüntülerin en/boy bilgisinin ayarlandığı bölüm olacaktır.

□ .bmp kaydetme: Parse işlemi sonrası tüm görüntü çerçevelerinin sadece Y bileşenleri numaralandırılarak kaydedilecektir.

□ Görüntü oynatma/gösterme: Parse işlemi sonrası sadece Y bileşenleri, tasarlanacak arayüz üzerinde

oynatılacaktır. Oynatma işlemi, çerçeve bazlı yapılacaktır.

Kısıtlamalar:

□ Sadece görüntü oynatma/gösterme ve görüntü kaydetme aşamalarında kütüphane dosyalarına başvurulabilir.

□ Kod geliştirme ortamı MATLAB, .net vb. olabilir.

□ Yazılım dili, C, C++, C# olacaktır.

1. Giriş

Öncelikle YUV çözücü uygulamamızda programlama dili olarak c#, ide olarak da visual studioyu kullandık bize birçok kolaylık sağladı. İlk olarak ödevi hiç anlamadık araştırma ödevi olduğu belliydi kodu kısa olmasına rağmen ilk 2 ödevden çok daha fazla zorladı. Saatlerce araştırdık arkadaşlarımıza sorduk yine olmadı en sonunda başladık. Başladıktan sonra da sorunlarla karşılaştık.

Öncelikle bizden arayüz tasarımı istendi.

Oluşturacağımız arayüzde dosya okuma, .bmp kaydetme, görüntü oynatma ve ayarlar seçeneği olması istenmiştir. Biz de buna uygun bir arayüz ayarladık.

Dosya okumaktan kasıt .yuv uzantılı dosyaları okuma ve parse işlemleri yapmaktır.

Ayarlar kısmında hangi formatta okuma işlemi yapılacağı ve fotoğraf penceresinin en boy bilgisi istenmektedir.

.bmp kaydetme: Parse işlemi sonrası tüm görüntü çerçevelerinin sadece Y bileşenleri numaralandırılarak kaydedilecektir.

Görüntü oynatma/gösterme: Parse işlemi sonrası sadece Y bileşenleri, tasarlanacak arayüz üzerinde oynatılacaktır. Oynatma işlemi, çerçeve bazlı yapılacaktır.

Programın çalışması ve tasarımı için bizden istenen Arayüz Tasarımı, Dosya Okuma, Ayarlar, .bmp kaydetme, oluşturduğumuz görüntüler için Görüntü oynatma ve gösterme işlemlerini bir bir gerçekleştirdik. Her adım için ayrı ayrı düşündük ve sırasıyla her bir adımı gerçekleştirdik. Programın daha en başında Arayüz

tasarımını ele aldık ve bu adımdan sonra diğer işlemlerde daha net bir şekilde ilerledik. Bu bölümde kullanıcıdan aldığımız inputlarla işleyeceğimiz resim dosyalarını, en ve boy bilgilerini almış olduk. Aldığımız bu dosyanın byte bilgilerini okuduk ve bir diziye kaydettik. Bu dizimiz üzerinden bizden istenen işlemleri gerçekleştirdik.

Ödevin büyük kısmı mantığını kavramaya çalışmakla ve ödevi anlamaya çalışmakla geçti. Uzun süren araştırmaların sonucunda mantığını anladıktan sonra da birçok sorunla karşılaştık sonunda projeyi başarıyla tamamladık. Bu proje bize görüntü işlemeyi ve .yuv uzantılı dosya bilgilerini almayı öğretti.

2. Temel Bilgiler

Visual Studio, Microsoft tarafından üretilen ve konsollar, grafik kullanıcı arayüzleri, Windows formları, Web servisleri ya da Web uygulamaları oluşturmak için kullanılan bir IDE'dir. Visual Studio programı içerisinde yalnızca Microsoft Windows tarafından desteklenen yerel kodlar kullanılmaktadır.

Visual Studio'nun bir nevi yazılım üretmek için yazılım olduğunu söyleyebiliriz. Visual Studio yazılımları elbette kendisi kodlamıyor ancak geliştiricilere ileri seviye özellikleri sayesinde yardımcı olarak daha kısa süre içerisinde programlarını hazırlama imkanı sunuyor. Visual Studio programını kullanarak bilgisayar yazılımları, web uygulamaları ve web servislerini çok daha hızlı ve kolay bir şekilde hazırlayabilirsiniz. Visual Studio son derece fonksiyonel ve şık ara yüz tasarımına sahiptir. Bu fonksiyonel arayüzle birlikte Visual Studio yazılımcıların yükünü hafifletecek bir kod editörü, debugger, GUI tasarlama aracı, veri tabanı şema tasarım aracı ve öncül revizyon kontrol sistemlerini sunuyor. Visual Studio yazılımın ticari amaçla kullanılan ticari program şeklinin yanı sıra topluluk bazlı olarak geliştirilen ücretsiz sürümü de bulunmaktadır.

Microsoft, yazılım geliştiricilere yansıyan sonuçları izliyordu ve yazılım geliştiricilerin bir takım sorunlar ile karşılaştıklarını bu sorunları çözmenin bir yolunu aramaktaydı.

Bu sorunları şu şekilde sıralamaktaydı:

Yazılan uygulamalara istedikleri an, kesintisiz ve güvenli bir şekilde ulaşamamak.

Uygulama geliştirirken yazılımcıların, altyapı kodlarını kodlamalarından dolayı geliştirme ve test süresinin uzaması.

Bir uygulamanın farklı platformlarda çalışması için ortak kodların bulunmaması.

Microsoft 1990 yılında bu tür sorunlara çözüm bulmak amacıyla yazılım geliştiricilerin sorunlarını çözecek bir platform geliştirmeye başladı. Bu platformun adı .NET platformuydu. Bu platform her türlü yazılım geliştirme ihtiyacına hazır bir altyapı sunmuştur. Böylece yazılımlar daha kolay geliştirilmeye başlandı.

.NET Framework Nedir ?

.NET Framework, bileşen yönelimli uygulamaların geliştirilmesini destekleyen bir ortamdır. Farklılık gösteren bilgisayar dillerinin bir arada çalışmasını mümkün hale getiren bir platformdur. Windows platformu için güvenlik, taşınabilirlik ve ortak bir programlama modeli sağlar.

.Net sınıf kütüphanesi de programlama amacı ile geliştirilmiştir. Bu kütüphane, programın çalışma ortamına erişmesine izin verir.

Common Language Runtime Nasıl Çalışır ?

CLR, .NET kodunun çalıştırılmasını sağlar. Bir derleyiciden elde edilen çıktı çalıştırılabilen bir kod değildir. Microsoft Intermediate Language (MSIL) denilen özel bir tipte dosyadır. Program çalıştığı zaman MSIL kodlarını çalıştırmak CLR'nin görevidir. CLR, .NET Framework'un taşınabilirliğinin bir parçasıdır.

Visual Studio her ne kadar bir geliştirme ortamı olsa dahi Eclipse gibi bu ortam içerisinde birçok programlama dilini kullanamıyor, sadece Visual Studio tarafından desteklenen diller üzerinden çalışabiliyorsunuz. Visual Studio yazılımının desteklediği programlama dilleriyse şunlardır;C, C++, C#, Visual Basic .NET, F#, Fossil, M, Python, HTML/XHTML/CSS, JavaScript.

YUV, genellikle video görüntülerini kaydetmek için kullanılan bir renk sistemidir. Y: Luminance, U: Chrominance1, V: Chrominance2 sözcüklerinin baş harflerinden oluşan kısaltmadır.

Sistemde Y işareti siyah-beyaz, U (Cb:Chrominance blue) ve V (Cr:Chrominance red) işaretleri ise mavi ve kırmızı renk bilgilerini temsil ederler. YUV birçok alt standarta sahiptir. Örnek bir YUV çevrimi olarak, RGB ile HDTV standartındaki YUV dönüşümü inceleyebiliriz: ITU.BT-709 HDTV YCbCr

$$\{ \displaystyle Y=0.2215*R+0.7154*G+0.0721*B \}$$

$$\{ \displaystyle Cb=-0.1145*R-0.3855*G+0.5000*B \}$$

$$\{ \displaystyle Cr=0.5016*R-0.4556*G-0.0459*B \}$$

3. Geliştirilen Mimari

Proje geliştirilmeye başlamadan önce projenin tasarımı aşamasında nasıl bir algoritma kurmamız gerektiğini düşündük. Ve doğru tercihin c# programlama üzerinde ilerlememiz gerektiğini bulduk. İlk olarak kullanıcıya hangi işlem yapılmasını öğrenmek için button kullanarak işlemin türünü sorduk. Kullanıcının istediği işleme uygun şekilde buttonların birbiri arasındaki etkileşimlerini de düşündük. Kullanıcının seçtiği rastgele matris oluşturma veya kullanıcının matris oluşturma seçeneklerini kullanıcının kullanabileceği şekilde bir ara yüz yardımıyla oluşturduk. Hatalara sebep olmaması için kullanıcı hatalarına önlem alındı hatalı yaptığı her giriş için onay tuşlarına bastığında message box kutularıyla yanlış girişleri uyarılarak doğru giriş yapmaları için kolaylıklar sağlandı. Kullanıcı kendi pencere bilgilerini oluşturma için textboxlara girdiği en boy bilgileri alındı ve bunlara uygun işlemler yapıldı. Textboxları kullanıcı isteğine göre dolduruldu ve o değerlerden işlemler yapıldı. Girilen pencere ve format bilgilerine göre işlemler yapıldı yapılan işlemler sonucunda .yuv uzantılı dosyanın y değerleri çekildi ve çekilen y değerlerine parse işlemleri uygulanarak görüntü gösterildi. III. dönem Nesneye Yönelik Programlama dersinde gördüğümüz class yapıları, Programlama derslerinde gördüğümüz temel algoritmalar bu projenin mimarisinin gelişmesinde rol oynadı.

3.1 Kullanılan Fonksiyonlar

Ödevin kodu kısa olduğundan dolayı metotlara fazla başvurmamak. Arayüz tasarımıındaki butonların oluşturduğu metotlar ve visual studionun desteklediği fonksiyonlar dışında bir şeye başvurmamak.

3.2 Kullanılan Veri Yapıları

Projemiz c# programlama ile nesneye yönelik programlamayı tam anlamıyla kavrayabilmek için verilmiştir. Bizde c# ile class yapıları üzerinde buton, Connection, Statement, ResultSet yapılarını kullanarak projeyi sonlandırmayı planladık. İstenen yapıları önce Ahmet Sayar hocamızın öğrettiği slaytlar ile çalışmalarımızı sürdürdük. Bu slaytların yanı sıra internet üzerinden de birçok araştırmamız oldu. Textboxlara bu projede çok fazla ihtiyacımız oldu onları görünürlüğü kullandık arayüz tasarlarken içlerine yazılan değerleri tutturduk. Butonlara da onaylama görevlerini verdik butonların kod kısmında birçok kodu onların bulunduğu eventlerde yaptık.

3.3 Projenin Yapımı Sırasında Karşılaşılan Sorunlar

Birçok problemle karşılaştık açıkçası ilk okuduğumuzda hiçbir şey anlamadık her adımında problem yaşadık zorlandık ama her birini aşarak problemleri çözdük ve ödevi tamamladık. İlk karşılaştığımız sorun ödevi anlamamaktı sonra önceki ödevlerde neatbeans kullandığımızdan ötürü ideye de biraz alışmaya çalıştık. Kullandığımız dil de farklıydı çok benzemesine rağmen garipsedik ama alıştık. Sonra adım adım gitmeye başladık ilk yaşadığımız problem .yuv dosyalarını işlem yapmak üzere okumaktı deneme imkanımız olmadığı için doğru olduğunu düşünüp devam ettik .yuv dosyasını okuduktan sonra dönüştürdük. Format mantığını anlamamıştık onu öğrendik ve y değerlerini ona göre çektik. Sonra deneye deneye matematiksel işlemleri yaptık ve ödevi tamamladık.

3.4 Yazılım Geliştirme İçin Harcanan Süreler

Proje anlatımı yapıldığı gün neler yapabiliriz hangi kütüphaneleri kullanabilir araştırmaya başladık. Benzer yapılmış uygulamaları tarayıp nasıl bir yol izlemeliyiz buna karar verip proje için çalışmaya başladık ve toplam 20 günde uygulama son halini aldı. Kişi başı yaklaşık çalışma saatleri:

Emre Altay: 90 saat, Batuhan Subaşı: 90 saat

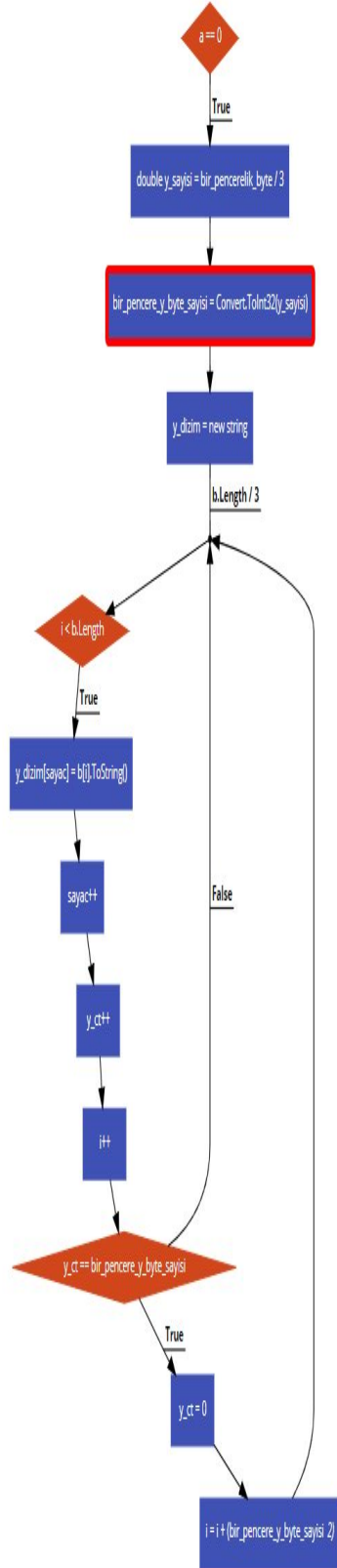
3.5 Problem Tanımı

Bir resim formatının farklı bir renk uzayında görüntülenmesi istenmektedir. Bizden dönüşüm işlemleri yaparak sadece y değerlerinden fotoğraf göstermektir.

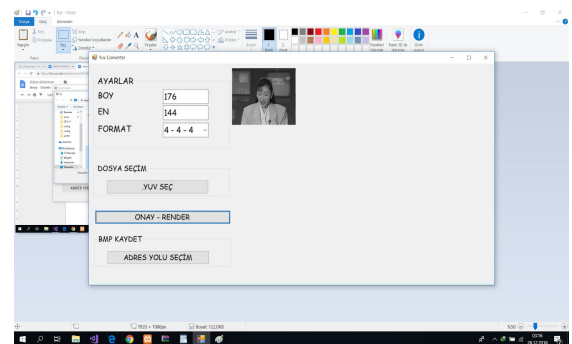
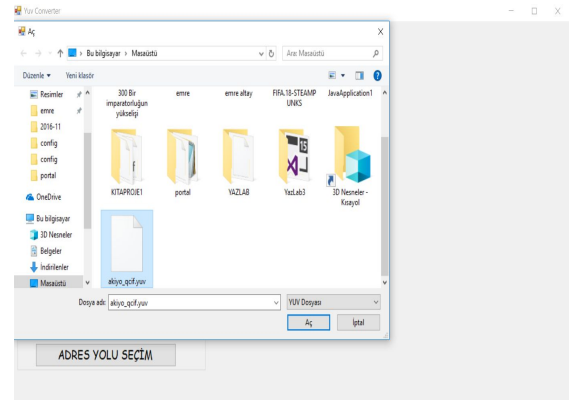
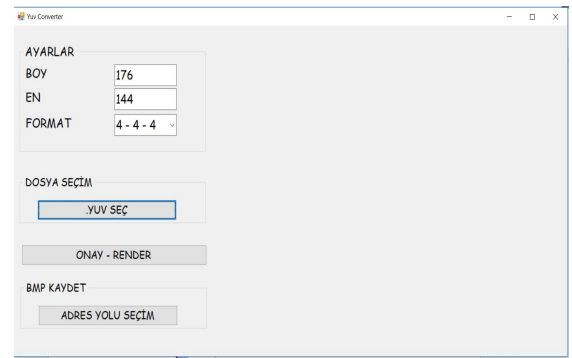
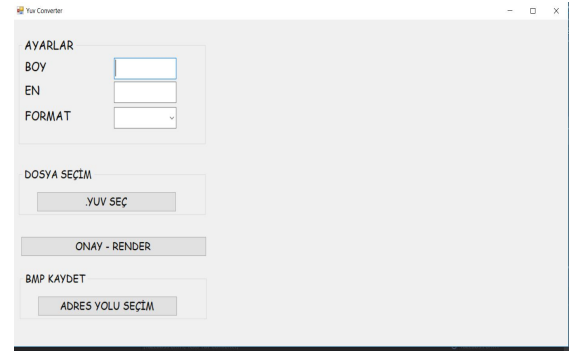
3.6 Yapılan Araştırmalar

Yapılan araştırmalar projenin ilk adımıyla ödev verilir verilmez yol yöntem belirlemek ve fikir sahibi olmak için araştırmalara başladık. İlk işimiz adım adım konuya yaklaşmak oldu. Araştırmalarımız adım adım devam etti kafamızdaki algoritmaya uygun kodu elde etmeye çalıştık sorunlar karşımıza çıktıkça araştırmaya devam ettik. Bilmediğimiz birçok terim vardı rgb, .yuv uzantılı dosyalar ve bmp gibi birçok terimi öğrendik.

3.7 Akış Şeması



4. Kullanıcı Kataloğu



5. Sonular

Sonu olarak devimiz kusursuz bir biimde alıřmaktadır. Bizden istenen btn isterler programda verilmektedir.

6. Kaynaka

- <https://tr.wikipedia.org/wiki/YUV>. [Ulařım Tarihi: Aralık 12, 2018].
- <https://www.kodyaz.net/c-tur-donusumleri/> [Aralık 18, 2018]
- <http://trace.eas.asu.edu/yuv/index.html> [Aralık 19, 2010]
- <http://samples.mplayerhq.hu/raw-video/squirrel-720x576-422P.yuv> [Aralık 19, 2008]
- <http://samples.mplayerhq.hu/raw-video/squirrel-720x576-444P.yuv> [Aralık 19, 2008]