KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

GERÇEK ZAMANLI GÖMÜLÜ ŞERİT TESPİT SİSTEMİ

RECEP TALHA KÜYÜK

KOCAELİ 2014

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ

ANABİLİM DALI

YÜKSEK LİSANS TEZİ

GERÇEK ZAMANLI GÖMÜLÜ ŞERİT TESPİT SİSTEMİ

RECEP TALHA KÜYÜK

Prof. Dr. Sarp ERTÜRK **Danışman, Kocaeli Üniv**	.....
Doç. Dr. Songül ALBAYRAK **Jüri Üyesi, Yıldız Teknik Üniv.**	.....
Yrd. Doç. Dr. Sultan Aldırmaz ÇOLAK **Jüri Üyesi, Kocaeli Üniv.**	.....

Tezin Savunulduğu Tarih: 29.12.2014

ÖNSÖZ VE TEŞEKKÜR

Öncelikle bu konuda çalışma yapmama imkan tanıyan, bu süreçte bana her türlü desteği ve motivasyonu sağlayan danışman hocam Prof. Dr. Sarp ERTÜRK'e en içten dileklerimde teşekkürlerimi sunarım.

Tez çalışmalarım boyunca beraber çalıştığımız ve yardımlarını esirgemeyen Mustafa ILGAR'a ayrıca teşekkür ederim. Bunun yanında çalışmamda bana yol gösteren ve motive eden TÜBİTAK iş arkadaşlarımdan başta Dr. İbrahim HÖKELEK ve Onur ERTÜRK olmak üzere diğer tüm arkadaşlarıma teşekkürü borç bilirim.

Ayrıca Yüksek Lisans Tez çalışmamın zor dönemlerinde hep yanımda olan ve hiçbir zaman desteklerini esirgemeyen Aileme minnettarım.

Aralık – 2014 Recep Talha KÜYÜK

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR i
İÇİNDEKİLER ii
ŞEKİLLER DİZİNİ iii
TABLOLAR DİZİNİ iv
SİMGELER ve KISALTMALAR v
ÖZET vi
ABSTRACT vii
GİRİŞ 1
1. ŞERİT TESPİT SİSTEMLERİ 3
1.1.Sobel Filtresi 3
1.1.1.Düşey yönde sobel filtresi 3
1.1.2.Yatay yönde sobel filtresi 5
1.2.1-BitDönüşümü 6
1.3.Çarpmasız 1-Bit Dönüşümü 7
2. SİSTEM TASARIMI 9
2.1.Donanım Platformu 9
2.1.1.Beagleboard-xM 10
2.1.2.DM3730 işlemcisi 11
2.1.2.1.ARM cortex-A8 işlemci ünitesi 12
2.1.2.2.CMEM 13
3. YAZILIM 15
3.1.Angström İşletim Sistemi 15
3.1.1.U-Boot konfigürasyonu 16
3.1.2.Kernel konfigürasyonu 17
3.2.DVSDK 17
3.2.1.DVSDK 4.01 kurulumu 17
3.2.2.DMAI 18
3.2.3.C6Accel 19
3.3.OpenCV 21
4. UYGULAMA YÖNTEMLERİ 22
4.1.ARM Tabanlı OpenCV Kullanımı 22
4.2.ARM Tabanlı OpenCV ve DSP Birlikte Kullanımı 26
4.3.ARM ve DSP Birlikte Kullanımı 29
4.4.Şerit Tespit Karar Algoritması 33
5. SONUÇLAR VE ÖNERİLER 36
KAYNAKLAR 37
KİŞİSEL YAYIN ve ESERLER 39
ÖZGEÇMİŞ 40

ŞEKİLLER DİZİNİ

Şekil 1.1. Düşey yönde Sobel Filtresi 4
Şekil 1.2. Örnek imge 4
Şekil 1.3. Düşey Sobel Filtresi uygulanmış imge 5
Şekil 1.4. Yatay yönde Sobel Filtresi 5
Şekil 1.5. Yatay yönde Sobel Filtresi uygulanmış imge 6
Şekil 1.6. Orijinal imge 7
Şekil 1.7. 1-Bit dönüşüm uygulanmış imge 7
Şekil 1.8. Çarpmasız 1-bit dönüşüm filtresi 8
Şekil 2.1. BeagleBoard-xM kart şeması 10
Şekil 2.2. DM3730 fonksiyonel blok diyagram 12
Şekil 2.3. ARM® Cortex®-A8 işlemci ARMv7 mimarisi 13
Şekil 3.1. Araç zinciri ile çapraz derleme 16
Şekil 3.2. DMAI blok diyagramı 18
Şekil 3.3. C6Accel SoC görünümü 20
Şekil 3.4. C6Accel kullanımı ve işlemci yükü karşılaştırması 20
Şekil 4.1. ARM tabanlı sistem mimarisi 22
Şekil 4.2. ARM üzerinde kenar bulma yöntemleri süre karşılaştırması 23
Şekil 4.3. ARM üzerinde MF-1BIT ve 1BIT dönüşümü karşılaştırması 24
Şekil 4.4. İmge karşılaştırmaları, Orjinal imge (a), Gri (b), 1BT, MF-1BT (d) 25
Şekil 4.5. MF-1BT çeşitli eşik değerleri için çıkış imgeleri 26
Şekil 4.6. ARM ve DSP (C6ACCEL) kullanan sistem mimarisi 27
Şekil 4.7. DSP'de C6Accel kullanarak kenar bulma süreleri karşılaştırması 28
Şekil 4.8. ARM ve DSP (DMAI ve C6ACCEL) kullanan sistem mimarisi 29
Şekil 4.9. YUYV piksel format 30
Şekil 4.10. DSP Üzerinde 1-BIT dönüşümü blok diyagramı 31
Şekil 4.11. DSP DMAI kullanarak kenar bulma süreleri karşılaştırması 32
Şekil 4.12. Şerit tespit karar algoritması diyagramı 34
Şekil 4.13. 1BT işlenmiş ve ROI belirlenmiş imge 34
Şekil 4.14. Araç şerit içinde durumu 35
Şekil 4.15. Araç sağ bölgeden şerit ihlali durumu 35

TABLOLAR DİZİNİ

Tablo 4.1. Sistem mimarisi-1 performans sonuçları. 25
Tablo 4.2. Sistem mimarisi-2 performans sonuçları. 29
Tablo 4.3. Sistem mimarisi-3 performans sonuçları 33

SİMGELER ve KISALTMALAR

Kısaltmalar

API	: Application Programming Interface (Uygulama Programı Arabirimi)
CMEM	: Contiguous Memory Management (Bitişik Bellek Yönetimi)
CODEC	: Coder-Decoder (Kodlayıcı-Kod Çözücü)
DMA	: Direct Memory Access (Direkt Bellek Erişimi)
DMAI	: Davinci Multimedya Uygulama Ara Yüzü
DSP	: DigitalSignalProcessor (Dijital Sinyal İşlemcisi)
DSPLIB	: DigitalSignalProcessor Library (Dijital Sinyal İşlemci Kütüphanesi)
DVI	: Digital Video Interface (Dijital Video Arayüzü)
DVSDK	: Digital Video Software Development Kit(Dijital Video Yazılım Geliştirme Kiti)
FPS	: Frameper Second (Saniye Başına Çerçeve)
FPU	: Floating Point Unit (Kayan Noktalı Ünite)
GCC	: GNU Compiler Collection (GNU Derleyici Koleksiyonu)
IMGLIB	: Image Library (Görüntü Kütüphanesi)
IPC	: InterprocessorCommunication (Süreçler Arası İletişim)
LCD	: Liquid CrystalDisplay (Sıvı Kristal Ekran)
LPM	: LocalPower Manager (Düşük Güç Yöneticisi)
MMC	: Multimedia Card (Multimedya Kart)
MMU	: Memory Management Unit (Bellek Yönetim Ünitesi)
OMAP	: Open Multimedia Application Platform(Açık Multimedya Uygulama Platformu)
OS	: Operating System (İşletim Sistemi)
POP	: Package on Package (Paket üstü Paket)
RAM	: Random Access Memory (Rasgele Bellek Erişimi)
RGB	: Red Green Blue (Kırmızı Yeşil Mavi)
RISC	: ReducedInstruction Set Computer(Azaltılmış Komut Takımı Bilgisayarı)
RTOS	: Real Time Operating System (Gerçek Zamanlı İşletim Sistemi)
ROI	: Region of Interest (İlgi Bölgesi)
ROM	: Read-Only Memory (Sadece Okunabilir Bellek)
SDMA	: System Direct Memory Access (Sistem Direkt Bellek Erişimi)
SDK	: Software Development Kit (Yazılım Geliştirme Kiti)
SIMD	: Single Instruction Multiple Data (Tek Komut Çoklu Veri)
SoC	: System on Chip (Çip üzeri Sistem)
TI	: Texas Instruments (Texas Ölçümleri)
USB	: Universal SerialBus (Evrensel Seri Yolu)
V4L2	: Video for Linux 2 (Linux İçin Video 2)
VISA	: Video, Image, Speech, Audio (Video, Görüntü, Konuşma, Ses)

GERÇEK ZAMANLI GÖMÜLÜ ŞERİT TESPİT SİSTEMİ

ÖZET

2012 yılında Türkiye'de gerçekleşen ölümlü ve yaralanmalı kazalarda insan faktörü % 99,12 gibi çok büyük oranla birinci derecede sorumlu görülmektedir. Sürücüye ait kusurlu kazalarda aşırı hız etkeni % 31,61 oranı ile başı çekerken doğrultu değiştirme kurallarına uymama % 14,90 oranı ile ikinci büyük kusur payı olmuştur. En önemli yol güvenliği uygulamalarından biri olan şerit takip sistemi konusunda pek çok araba üretici firmaları çalışmalar yapmaktadır.

Bu tezin konusu yol güvenliği uygulamalarından biri olan şerit tespit sisteminin gerçek zamanlı gömülü sistemlerde gerçeklemesi ve performans analizinin yapılması üzerinedir. Çalışmamızda mobil ortamlarda yeterli görüntü kaydetme ve işleme kabiliyetine sahip, düşük güç tüketen ve kolay uygulama geliştirme ortamı sağlayan TI DM3730 System on Chip (SoC) bulunan BeagleBoard-xM platformu kullanılmaktadır. DM3730 SoC içerisinde ARM işlemcisinin yanısıra DSP çekirdeğinin olması avantajıyla şerit tespit uygulaması için üç farklı sistem mimarisini gerçekledik. Birinci mimaride görüntüyü almak, işlemek ve anlamlandırmak için sadece ARM işlemcisi üzerinde çalışan OpenCV kütüphanesi kullanılırken diğer iki mimaride ise ARM işlemcisi ve DSP birlikte kullanılmaktadır.

Şerit tespit sisteminin en önemli bileşenlerinden birisi kenar bulma algoritmasıdır. Bu tezde kenar bulma algoritması olarak literatürde yaygın olarak kullanılan sobel filtresinin yanı sıra 1-bit dönüşümü ve çarpmasız 1-bit dönüşümü olmak üzere üç farklı yöntem gerçeklenmiştir. Bu üç farklı yöntemin yukarıda belirtilen üç farklı mimari için performans karşılaştırmaları yapılmıştır.

Anahtar Kelimeler: ARM ve DSP Mimarisi, BeagleBoard-Xm Performans Analizi, Gerçek Zamanlı Görüntü İşleme, Gömülü Sistem Dizayn, Şerit Tespit Sistemi.

REAL TIME EMBEDDED LANE DETECTION SYSTEM

ABSTRACT

The human factor has been reported as the main reason of traffic accidents for Turkey in 2012 such that 99.12% of the traffic accidents which resulted in death or injury have been caused by the humans. While 31.61% of the accidents caused by the driver were on account of excessive speed, an improper and unsafe lane change has been in the second place with 14,90%. Lane tracking systems have become one of the major road safety applications and have been extensively studied by many industry-leading car manufacturers.

The main topic of this thesis is the implementation and realization of a lane tracking application in an embedded system and the performance analysis in real time. In this perspective, we used the BeagleBoard-xM platform which carries a TI DM3730 System on Chip (SoC) that enables the easy development of mobile video capture and signal processing applications with low battery usage in mobile devices. Since DM3730 SoC includes a DSP core in addition to the ARM processor, we developed and implemented three different system architectures for the lane tracking application. While the first architecture uses OpenCV library that runs on only ARM processor, in order to capture, process and make use of images, the other two architectures use both ARM and DSP processors.

One of the main components of the lane tracking system is the edge detection algorithm. In addition to the Sobel algorithm, which is widely used in literature for this purpose, we implemented 1-bit transform and multiplication free 1-bit transform algorithms for a fast edge detection computation. A performance analysis and comparison has been made for these three algorithms using the above mentioned three system architectures.

KeyWords: ARM and DSP Architecture, BeagleBoard-Xm Performance Analyze, Real Time Image Processing, Embedded System Design, Lane detection.

GİRİŞ

Günümüzde teknoloji çok ilerlemesine rağmen, yol güvenliğinin sağlanması konusunda gelinen nokta yetersiz kalmaktadır. 2012 yılında Türkiye'de gerçekleşen ölümlü ve yaralanmalı kazalarda insan faktörü %99,12 gibi çok büyük oranla birinci derecede sorumlu görülmektedir. Sürücüye ait kusurlu kazalarda aşırı hız etkeni %31,61 oranı ile başı çekerken doğrultu değiştirme kurallarına uymama %14,90 oranı ile ikinci büyük kusur payı olmuştur [1].

Google firmasının "Google Self-Driving Car" isimli projesinde otonom araç teknolojisini geliştirdiği ve gerçek yol ortamında test ettiği bilinmektedir [2]. Bu tür insansız araçların otomatik olarak çalışan yol güvenliği uygulamaları kullanması zorunludur. Şerit takip uygulaması bu sistemin en önemli yol güvenliği uygulamalarından biridir. Karayolu taşımacılığı terminolojisinde Şeritten Ayrılma Uyarı Sistemi (LDWS) araç şeritten ayrılmaya başladığında (o yönde dönüş sinyali vermediğinde) sürücüye uyarı vermektedir. LDWS'in daha ileri versiyonunda aracı otomatik olarak şerit sınırlarının içerisinde tutan Şerit İçerisinde Tutma Yardımcısı (LaneKeepingAssistSystem-LKAS) özelliği bulunmaktadır. LDWS ve LKAS sistemlerinin Audi, Citroen, Mercedes Benz, General Motors, Toyota firmalarının yeni nesil araçlarında kullanılmaya başlanmaktadır. Bununla birlikte diğer pek çok markada henüz böyle bir sistem yoktur.

Bu konuyla ilgili olarak pek çok akademik çalışma bulunmakla birlikte bu çalışmaların araçlara uygulanması sınırlı kalmıştır. Araştırma boyutundan bakıldığında, bunun ana nedenlerinden biri akademik çalışmaların gerçek zamanlı gömülü sistemler yerine daha çok bilgisayar ortamında yapılıyor olmasıdır. Gömülü sistemler kapasiteleri daha sınırlı olmakla birlikte önemli derecede fiyat ve güç tüketimi avantajları vardır. Bilgisayar ortamında geliştirilen yol güvenliği algoritmalarının gömülü sistemlerde çalıştırılmasıyla birlikte araçlarda yaygın olarak kullanılması beklenmektedir.

Bu tez yol güvenliği uygulamalarından biri olan şerit tespit sisteminin gerçek zamanlı gömülü sistemlerde gerçeklemesini kapsamaktadır. Bu tezde mobil ortamlarda yeterli görüntü kaydetme ve işleme kabiliyetine sahip, düşük güç tüketen ve kolay uygulama geliştirme ortamı sağlayan TI DM3730 işlemcisi seçilmiştir. DM3730 yongası üzerinde işlemci olarak 1-GHz ARM® Cortex™-A8 çekirdeği ve sinyal işleme ünitesi olarak 800-MHz TMS320C64x+TM DSP çekirdeği bulunmaktadır. Uygulama donanım platformu için DM3730 yongasını barındıran ve çevresel ünitelerini de etkin bir şekilde kullanıma sunan BeagleBoard-xM geliştirme kartı seçilmiş ve SoC mimarisi incelenmiştir.

BeagleBoard platformunun kullanıldığı şerit tespit uygulaması için üç farklı sistem mimarisi kullanılabilir. Birinci mimaride görüntüyü almak, işlemek ve karar vermek için sadece ARM işlemcisi üzerinde çalışan OpenCV kütüphanesi kullanılır. İkinci ve üçüncü mimarilerde ise ARM işlemcisi ve DSP birlikte kullanılmaktadır. İkinci mimaride görüntüyü almak için OpenCV Kütüphanesi kullanılırken görüntüyü işlemek için DSP çekirdeğinden faydalanılmaktadır. Üçüncü mimaride ise OpenCV yerine TI firmasının görüntü alma ve işleme uygulamaları için sağladığı DVSDK Kütüphanesi DSP çekirdeği ile birlikte kullanılmaktadır. Bu mimaride ARM ve DSP arasındaki iletişim DVSDK içerisinde bulunan DMAI ara yüzü ile sağlanır.

Gerçekleme için sadece ARM işlemcisi kullanıldığında işletim sisteminin sağladığı avantajlardan yararlanılırken gerçek zamanlı uygulamanın gerektirdiği zaman ölçütleri sağlanamayabilir. Yüksek işlemci yükü gerektiren multimedya işlemleri DSP üzerinde gerçekleştirildiğinde zaman ölçütlerinin karşılanması kolaylaşır. Görüntünün saklanmasını ve iletilmesini gerektiren uygulamalar için DMAI ara yüzünün sunduğu kod çözücü motoru diğer iki mimariye göre önemli avantajlar sağlamaktadır. Bu tezde bu üç farklı mimarinin karşılaştırmalı performans analizi şerit tespit sistemi uygulaması için yapılmaktadır.

Şerit tespit sisteminin en önemli bileşenlerinden birisi kenar bulma algoritmasıdır. Bu tezde kenar bulma algoritması olarak literatürde yaygın olarak kullanılan Sobel filtresinin yanı sıra 1-bit dönüşümü ve çarpmasız 1-bit dönüşümü olmak üzere üç farklı yöntem gerçeklenmiştir. Bu üç farklı yöntemin yukarıda belirtilen üç farklı mimari için performans karşılaştırmaları yapılmıştır. Sobel filtre işlemini DSP de

gerçeklemek için C6Accel ara yüzü ile IMGLIB kütüphanesinde bulunan IMG_Sobel_3x3_8() fonksiyonu kullanılmıştır. 1-bit dönüşüm yöntemi için C6Accel'de yeni çekirdek tanımlanmış ve ARM tarafından çağrılmak üzere IMG_1BIT_8() fonksiyonu oluşturulmuştur.

1. ŞERİT TESPİT SİSTEMLERİ

Kenar bulma görüntü işlemede temel öneme sahip bir problemdir ve görüntüdeki keskin süreksizlikleri tanımlama ve yerini saptamaya dayanır. Kenar bulma imgede gereksiz bilgi ve verileri önemli ölçüde azaltırken önemli yapısal özellikleri korur. Nesne tespit işleminde kenar bulma ön planda olduğundan bu tezde kullanılan şerit tespit sisteminde önemli yere sahiptir. Literatürde her biri belirli kenar türlerine duyarlı olacak şekilde tasarlanmış birçok kenar bulma operatörü bulunmaktadır. Bunların seçiminde öne çıkan değişkenler kenar yönü, kenar yapısı, ortam gürültüsüdür. Operatörün geometrik yapısı hangi yöndeki kenarlara daha duyarlı olacağını belirler. Gürültülü imgelerde kenar bulma daha zordur çünkü gürültü ve kenarlar yüksek frekans bileşen içerirler [3].

1.1. Sobel Filtresi

Sobel operatörü olarak da bilinen Sobel filtresi imge işleme ve bilgisayar görüsünde özellikle kenar bulma algoritmalarında kullanılır ve imgeden kenarları ve geçişleri vurgulu olan yeni bir imge yaratır. IrwinSobel tarafından Stanford Yapay Zeka Projesinde 1968 yılında "İzotropik 3x3 İmge Gradyan Operatörü" olarak sunulmuştur. Sobel operatörü ayrık türev operatörüdür ve düşey ve yatay yönlerde maskeleme işlemi yaparak kenar buldurur. Sobel operatörünün benzerlerine göre en önemli farkı iki maskesinin katsayısı da sabit değildir ve istenilenlere göre türev operatörlerinin uygunluğunu bozmadan düzeltilebilirler.

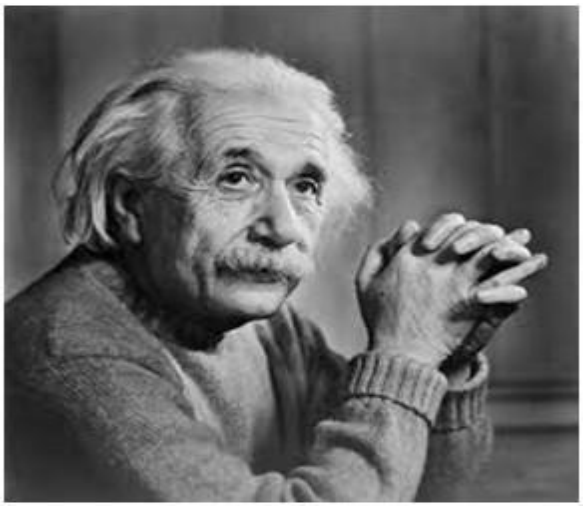
1.1.1. Düşey yönde sobel filtresi

Şekil 1.1'de görüldüğü üzere '0' lardan oluşan düşey yönde bir sütun içermektedir. Bu maske herhangi imge ile evirişim yapıldığında imgenin düşey kenarlarını ortaya çıkaracaktır.

-1	0	+1
-2	0	+2
-1	0	+1

Şekil 1.1. Düşey yönde Sobel Filtresi

Basitçe birinci dereceden türev mantığıyla çalışır ve kenar bölgesindeki piksel yoğunlukları farkını hesaplar. Merkez sütunun '0' olmasından dolayı imgenin orijinal değerlerini içermeyip bunun yerine kenarın etrafındaki sağ ve sol piksellerin yoğunlukları arasındaki farkı hesaplar. Birinci ve üçüncü sütunların değerleri 2 ve -2 olduğundan kenar bölgesi etrafındaki piksellere daha fazla ağırlık vererek kenar yoğunluğunu artırır ve orijinal imgeyle karşılaştırıldığında daha artırılmış hale gelir.



Şekil 1.2.Örnek imge



Şekil 1.3. Düşey Sobel Filtresi uygulanmış imge

1.1.2. Yatay yönde sobel filtresi

-1	0	+1
-2	0	+2
-1	0	+1

Şekil 1.4. Yatay yönde Sobel Filtresi

Şekil 1.4'te görüldüğü üzere '0'lardan oluşan yatay yönde bir sütun içermektedir. Bu maske herhangi imge ile evrişim edildiğinde imgenin yatay kenarlarını ortaya çıkaracaktır. Filtreleme işlemi bu maske temelinde ve belirli bir kenarın piksel yoğunlukları arasındaki farkı hesaplama mantığında çalışır. Maskenin merkez sütunu '0'lardan oluştuğu için imgedeki kenarın orijinal değerlerini içermez ancak o bölgenin altında ve üstündeki piksellerin yoğunlukları farkını hesaplar. Böylece yoğunlukların ani değişiklikleri artar ve kenar daha görülür hale gelir.



Şekil 1.5. Yatay yönde Sobel Filtresi uygulanmış imge

1.2. 1-Bit Dönüşümü

1-bit dönüşümü (1BT) orijinal imge ile bunun 17x17 boyutunda çoklu bant geçiren bir süzgeçten geçirilmiş hali karşılaştırılarak imgelerin piksel başına bir bit ile ifade edildiği bir yöntemdir. [4]'te hareket kestirimi için 1BT yöntemi uygulanmış ve 17x17 boyutlu çoklu bant geçiren bir süzgecin formu,

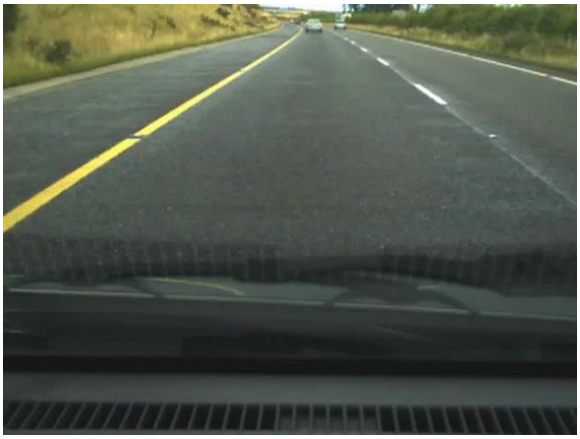
$$K(i,j)=\begin{Bmatrix}\frac{1}{25}, & i,j\in[0,4,8,12,16]\text{ise}\\ 0, & \text{diğer}\end{Bmatrix} \tag{1.1}$$

şeklinde ifade edilmiştir. Uygulanan filtrenin ardından Denklem (1.2) ile 1BT imge oluşturulmuştur. Denklem,

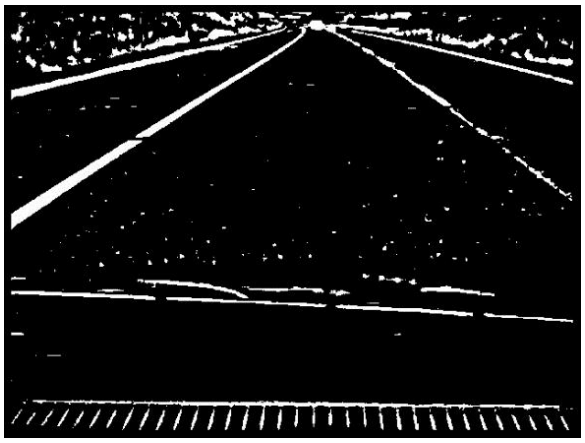
$$B(i,j)=\begin{Bmatrix}1, & \text{if } I(i,j)\geq I_F(i,j)\\ 0, & \text{diğer durumda}\end{Bmatrix} \tag{1.2}$$

şeklinde ifade edilmiştir.Burada *I(i,j)* orijinal imgeyi, *IF(i,j)* filtreden geçirilmiş imgeyi, *i* ve *j* ise piksel indislerini ifade etmektedir.

Burada Şekil 1.6'daki orijinal imge süzgeç ile filtrelenerek kenar bulma işlemi gerçekleşmiştir ve Şekil 1.7'deki 1-bit dönüşüm uygulanmış imge elde edilmiştir.



Şekil 1.6. Orijinal imge



Şekil 1.7. 1-Bit dönüşüm uygulanmış imge

1.3. Çarpmasız 1-Bit Dönüşümü

Çarpmasız 1-bit dönüşümünde temel mantık uygulanan filtrenin yapısını bozmadan "0" olmayan değerlerin 2'nin katı olmasıdır [5]. Bu sayede donanımsal olarak daha hızlı gerçekleşebilen işlem elde edilmektedir. Çarpmasız 1-bit dönüşümü temelinde 16 adet toplama ve 4 bit kaydırma işlemi yaptığından büyük ölçüde performans artırımı sağlamaktadır. Uygulanan filtre Şekil 1.8'de gösterilmiştir.

$$K = \frac{1}{16}\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Şekil 1.8. Çarpmasız 1-bit dönüşüm filtresi

Çarpmasız1-bit dönüşümü için kullanılan filtre, görüldüğü üzere 16 adet karo biçiminde yerleştirilmiş 1'lerden geri kalanı ise 0'lardan oluşmaktadır. Filtreleme işlemlerinin mantığı, imge üzerinde süzgecin blok-blok gezdirilmesiyle imge üzerindeki piksellere karşılık gelen değerlerle çarpılıp toplanarak, merkez piksele yeni değer olarak verilmesidir. Burada yapılan çarpma işlemi donanımsal olarak toplamaya karşılık daha büyük işlem yükü getirir. Kullanılan filtre bahsedildiği gibi yalnızca 1 ve 0 içerdiğinden herhangi bir çarpma işlemi yapılmadan yalnızca karşılık gelen piksellerin toplanıp 16'ya bölünmesi yani 4 bit kaydırılması işlemi yapılmıştır. Böylece donanımsal olarak büyük performans kazancı sağlanmıştır.

2. SİSTEM TASARIMI

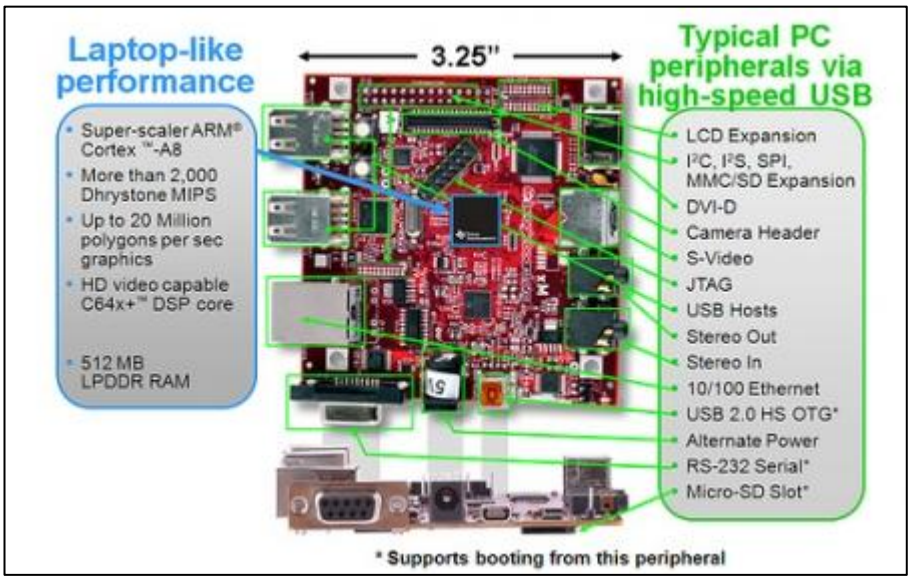
Bu tezde mobil ortamlarda yeterli görüntü kaydetme ve işleme kabiliyetine sahip, düşük güç tüketen TI DM3730 işlemcisini üzerinde barındıran ve kolay uygulama geliştirme ortamı sağlayan Beagleboard-xM gömülü platformu kullanılmaktadır. DM3730 yongası üzerinde işlemci olarak 1-GHz ARM® Cortex™-A8 çekirdeği ve sinyal işleme ünitesi olarak 800-MHz TMS320C64x+TM DSP çekirdeği bulunmaktadır. ARM Cortex A-8 çekirdeği üzerinde Angström Linux işletim sistemi çalıştırılmaktadır. TI DVSDK geliştirme ortamı yazılımı yardımıyla DSP üzerinde DSP/BIOS gerçek zamanlı çekirdek kullanılmaktadır. Bu sayede yüksek işlemci gerektiren görüntü işleme görevleri ARM işlemcisini kullanmadan DSP üzerinde daha hızlı gerçeklenmektedir. Bu mimari en iyi ARM işlemci gücünü ve grafik işleme yeteneğini düşük güç tüketimiyle sunar. Bu tezde üç farklı sistem mimarisi kullanılmıştır. Birinci mimaride görüntüyü almak, işlemek ve karar vermek için sadece ARM işlemcisi üzerinde çalışan OpenCV kütüphanesi kullanılmaktadır. İkinci ve üçüncü mimarilerde ise ARM işlemcisi ve DSP birlikte kullanılmaktadır. Aşağıdaki bölümlerde bu üç mimaride kullanılan birimler daha detaylı anlatılmaktadır. Ayrıca bu çözümlerin karşılaştırmalı performans analizi yapılmaktadır.

2.1. Donanım Platformu

Bu tezde mobil ortamlarda yeterli görüntü kaydetme ve işleme kabiliyetine sahip, düşük güç tüketen ve kolay uygulama geliştirme ortamı sağlayan TI DM3730 işlemcisi seçilmiştir. DM3730 yongası üzerinde işlemci olarak 1-GHz ARM® Cortex™-A8 çekirdeği ve sinyal işleme ünitesi olarak 800-MHz TMS320C64x+TM DSP çekirdeği bulunmaktadır. Uygulama donanım platformu için DM3730 yongasını barındıran ve çevresel ünitelerini de etkin bir şekilde kullanıma sunan BeagleBoard-xM seçilmiştir. Bu bölümde BeagleBoard-xM platformu çevresel üniteleri ve DM3730 mimari yapısı hakkında bilgi verilecektir.

2.1.1. Beagleboard-xM

Beagleboard özellikle açık kaynak topluluğu için üretilmiştir. DM3730 işlemcisinin tüm ara yüzlerini kullanıma sunmasa da standart ara yüzleri genişletilebilir olması birçok özellik eklenmesine olanak sağlar. Beagleboard son ürün tasarım öncesi test donanım platformu olarak kullanılabilir ve dizayn bilgileri ücretsiz olarak elde edilebilir [6]. Beagleboard içerisinde DM3730 1GHz işlemcisi 512MB RAM ile POP paketlenmiş olarak sunulmuştur. POP (Package on Package) tekniğinde RAM işlemcinin üzerine monte edilerek işlemcinin kapladığı alandan tasarruf sağlanır.



Şekil 2.1. BeagleBoard-xM kart şeması[7]

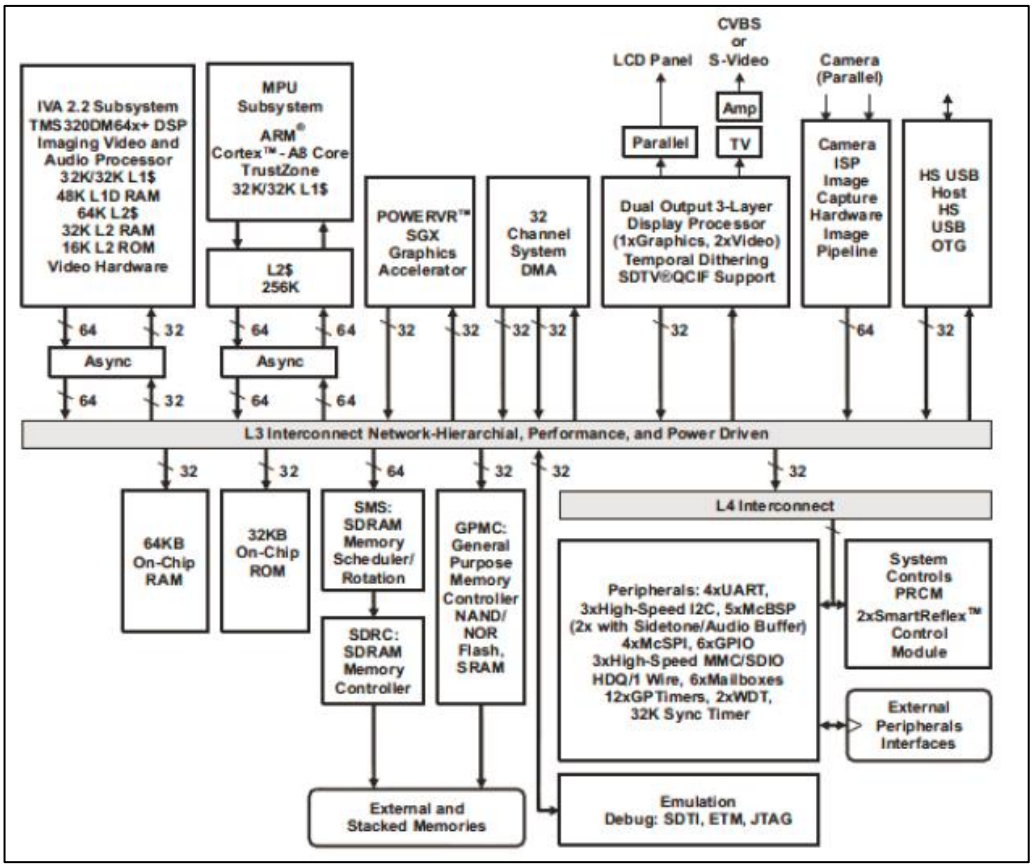
Beagleboard-xM'deönceki nesillerin aksine kalıcı NANDFlash bulunmamaktadır. Bu nedenle MMC girişinden SD kart kullanarak yada USB üzerinden depolama aygıtı kullanarak sisteme kalıcı eklenebilir. Kartta 4 adet USB girişi, Ethernet portu, DVI çıkışı, MMC mikro SD kart girişi, S-Video girişi, stereo ses girişi ve çıkışı, paralel port dijital kamera girişi gibi tipik bilgisayar çevre birimleri bulunmaktadır. Kartta güç yönetimi için TPS65950 entegresi kullanılmaktadır. TPS65950 entegresinde stereo ses giriş ve çıkışları kod çözücüleri, USB OTG fiziksel katmanları bulunmakla beraber tüm çevre birimlerin ve DM3730 işlemcisinin beslemesini sağlamak ve kontrol etmekle yükümlüdür.

2.1.2. DM3730 işlemcisi

DM3730 TI firmasının OMAP3530 işlemci ailesine mensup ve yüksek performansta dijital veri işleme kapasitesine sahip bir yongadır. OMAP işlemcileri özellikle taşınabilir cihazlar için gömülü multimedya uygulamaları için tasarlanmıştır. ARM CortexA-8 çekirdeği ile işletim sistemi çalıştırılabilir ve TMS320DM64x+ DSP sayesinde yüksek hızda dijital verileri işleyebilir. Bu mimari en iyi ARM işlemci gücünü ve grafik işleme yeteneğini düşük güç tüketimiyle sunar ve işlemci gücü 3D oyunlar, video konferans ve yüksek çözünürlüklü görüntü işleme uygulamaları için yeterlidir. Performans ve gücü çok iyi bir denge ile sunduğundan mobil cihazlar başta olmak üzere cep telefonları, navigasyon cihazları, medikal cihazları, ev otomasyonu ve test ve ölçüm cihazlarında kullanılmaktadır [8]. DM3730'u bir araya getiren alt sistemler şunlardır.

- ARM® Cortex™-A8 işlemci ünitesi MPU (1-Ghz'e kadar)
- TMS320C64x+TM DSP (800-MHz'e kadar)
- POWERVR SGX™ 3D grafik hızlandırıcısı
- Yüksek hızda veri transferi için L3 ve L4 bellekleri arabağlantısı

DM3730 işlemcisinin fonksiyonel blok diyagramı Şekil 2.2'de gösterilmiştir. L3 belleği arabağlantısı sayesinde ARM işlemcisi ve DSP sinyal işleme ünitesi arasında yüksek hızda veri aktarımı sayesinde iki donanım da aynı bellek üzerinden paralel işlem yapabilir. Ayrıca DM3730'da kullanılan SmartReflex™ 2 teknolojisi ile dinamik olarak boşta olan bellek bloklarının gücü kesilir ve transistorların akım kaçakları önlenerek performansları kontrol edilir [9]. Bu sayede OMAP3730 işlemcilerinde düşük güç tüketimi sunulmaktadır.



Şekil 2.2. DM3730 fonksiyonel blok diyagram

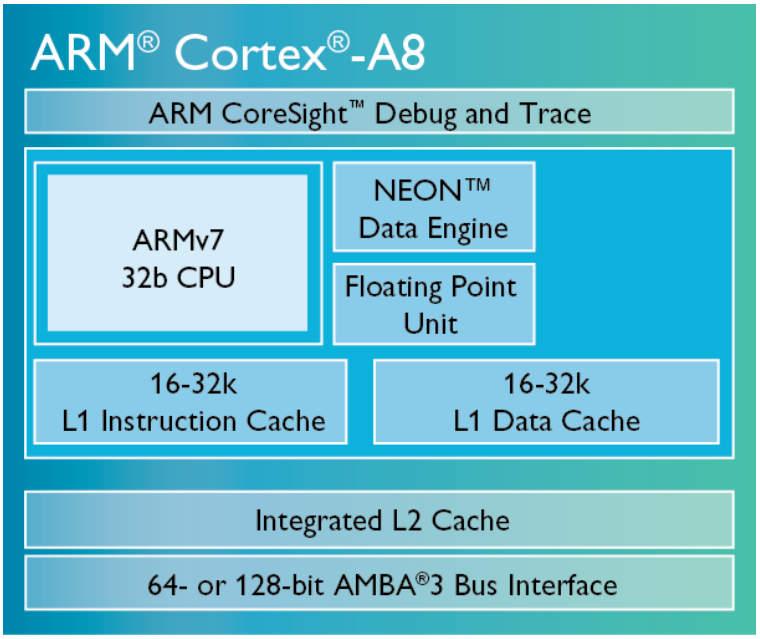
2.1.2.1. ARM cortex-A8 işlemci ünitesi

ARM Holdings tarafından geliştirilen Cortex-A8 işlemcisi ARMV7-A ve Azaltılmış Komut Takımı Bilgisayarı (RISC) tasarımına dayanır. Cortex-A8 yüksek performans ve düşük güç tüketimi gerektiren taşınabilir cihazlarda kullanmak üzere tasarlanmış tek çekirdekli bir işlemcidir.Cortex-A8'in öne çıkan başlıca özellikleri şunlardır.

- 600Mhz ile 1,5Ghz frekansları arasında çalışabilme
- Bellek yönetim sistemi (MMU)
- ARM Version 7™ ISA komut seti
- NEON™ SIMD, VFPv3 Floating Point, Thumb®-2 teknolojileri
- Level 1 32KB komut ve veri önbellek
- Level 2 1MB'a kadar ayarlanabilir hata düzeltme (ECC) destekli önbellek

Cortex-A8 ayarlanabilir çalışma frekansı seçeneği ile belirli bir frekansta çalıştırılabilir yada otomatik çalışma frekansı seçeneği aktive edilerek işlem yüküne göre en uygun frekansta çalışır. Bellek yönetim sistemi (MMU) Cortex-A8'in çeşitli

zengin işletim sistemleri uygulamalarıyla çalışabilme olanağı sağlar. Thumb®-2 teknolojisi geleneksel ARM kodları komutlarını saklarken %30'a kadar azalma sağlarken maksimum performans sunar. Şekil 3'te ARM Cortex-A8 işlemci ARMv7 mimarisi gösterilmiştir.



Şekil 2.3. ARM® Cortex®-A8 işlemci ARMv7 mimarisi[10]

NEON teknolojisi128-bit SIMD motoru (Tek Komut, Çoklu Veri) ARM Cortex-A serisi işlemciler için tasarlanmış ve tüketiciye gömülü sistemlerde multimedya uygulamaları için esnek, güçlü ve yüksek performanslı bir deneyim sunar. MPEG-4, H.264, JPEG, MP3, AAC gibi video ve ses kodlayıcı sıkıştırma ve çözme işlemlerinde, 2D/3D grafiklerinde ve görüntü işleme süreçlerinde %75'e varan performans artışı sağlar [11].

2.1.2.2. CMEM

ARM işlemci ve DSP veri alışverişini CMEM modülü ile tanımlanan ortak DDR2 bellek üzerinden yaparlar. CMEM bir veya biden fazla bitişik belleği yönetmeye yarayan uygulama programlama arayüzüdür (API). ARM bellekteki adreslerini MMU üzerinden sanal adresler olarak tanır. Fakat DSP'de MMU olmadığı için bellek adreslerini direk fiziksel olarak görür. Bu karmaşayı gidermek ve birlikte çalışabilmek için CMEM bellek adres dönüşümü yapmaktadır [12]. Bellek havuzu

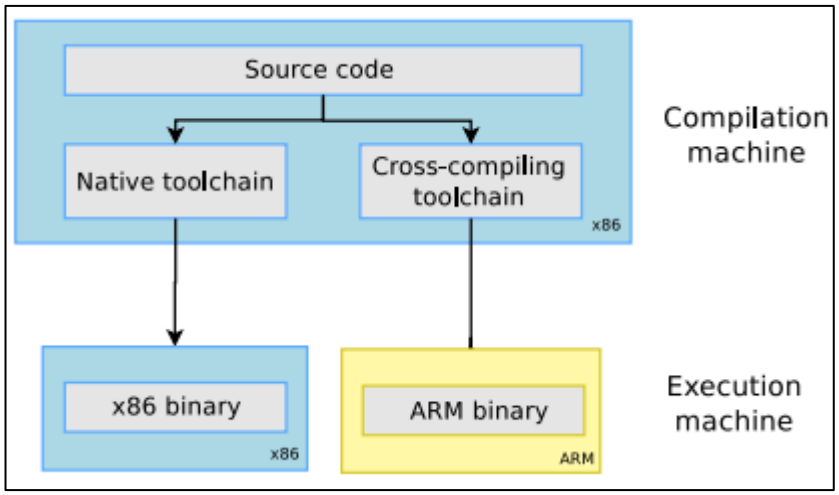
yapısıyla bellek parçalanmasını önler ve büyük bitişik bellek bloklarının kullanımını garanti altına alır.

3. YAZILIM

OMAP işlemcileri Linux®, Microsoft's Windows, Android™, Symbian OS™ ve Mobile™ işletim sistemlerini desteklemektedir. Bu tezde ARM işlemcisinde çalıştırılmak üzere Angström Linux dağıtımı seçilmiştir. DSP çekirdeğinin üzerinde çalışması için tekseçenek olan DSP/BIOS Real-Time işletim sistemi mevcuttur. ARM işlemcisi tüm çevre birim sistemlerini ve I/O giriş çıkışlarını kontrol ederken aynı zamanda DSP çekirdeği ile haberleşme ve koordinasyon görevini de üstlenir. Tüm haberleşme ortak bellek üzerinden (SDMA) okuma yöntemiyle hızlı bir şekilde yapılmaktadır.

3.1. Angström İşletim Sistemi

Angström birçok gömülü cihazları destekleyen bir Linux dağıtımıdır. Angström OpenZaurus, OpenEmbedded ve OpenSIMpad projelerinde çalışan küçük bir geliştirici grubu tarafından başlamış ve gömülü sistemler için daha kararlı ve kullanıcı dostu bir dağıtım ortaya çıkarmak için bu projelerin birleşimiyle oluşmuştur. OpenEmbedded Angström tarafından kullanılır ancak Angström bir OpenEmbedded değildir. OpenEmbedded çapraz derleme vasıtasıyla proje paketleri oluşturmak için bittake kullanır. Çapraz derleme yordam geliştirme ortamı ve gömülü sistem platformları farklı olduğunda kullanılır. Bu sayede geliştirme ortamı bilgisayarının işlemci gücü ve bellek kapasitesi avantajıyla gömülü sistem platformu için proje paketleri daha hızlı derlenir. Şekil 3.1'de araç zinciri (Toolchain) ile x86 tabanlı bilgisayarda ARM işlemci için çapraz derleme gösterilmiştir.



Şekil 3.1. Araç zinciri ile çapraz derleme

3.1.1. U-Boot konfigürasyonu

OMAP işlemcilerinde önyükleme 4 basamaktan oluşur;

- ROM kodu: İşlemcinin 64KB limitli ROM kısmında bulunan kod RAM'e yüklenir ve amacı geçerli bir önyükleme önyükleyicisi bulmaktır. Kullanıcıdan bağımsızdır.
- X-Loader (MLO): SRAM'den çalışır. DRAM, NAND yada MMC sürücüsünü yükler ve u-boot imajını bulup RAM'e aktarır. Kullanıcıdan bağımsızdır.
- U-boot: RAM'den çalışır. Network,USB gibi donanımları yükler ve NAND,MMC yada network üzerinden kernel imajını RAM'e yükler.
- Kernel: RAM'den çalışır ve tüm sistemi ayağa kaldırır.

Beagleboard-xM'de NAND Flash bulunmadığından önyükleme ve tüm işlemler MMC yuvasına takılan mikro SD kart ile yapılır. DM3730 işlemcisi fabrika çıkışlı ROM kodu MLO dosyasını bulmak üzere programlanmıştır. MLO ise u-boot.bin dosyasını bularak RAM'e aktarır ve çalışmasını sağlar. U-boot parametreleri ile seri-port ayarları, network cihazı, görüntü formatı ve ekran seçimi gibi özellikler ayarlanabilir. Bu tezde ekrana görüntü omapframebuffer overlay üzerinden gönderileceği için u-boot parametrelerinde,

setenvvram 14M omapfb.vram=0:6M,1:4M,2:4M

komutu gereklidir.Kullanılan u-boot parametreleri şu şekildedir. Bu komut ile overlay0 için 6 MB, overlay1 için 4MB ve overlay2 için 4MB yer ayrılmaktadır.

u-bootparametleri:

setenvbootargs 'console=ttyS2,115200n8 root=/dev/mmcblk0p2 rw ip=offmem=55M@0x80000000 mpurate=1000 omap_vout.vid1_static_vrfb_alloc=y omapfb.vram=0:6M,1:4M,2:4Mmem=384M@0x88000000 omapfb.mode=dvi:hd720 omapdss.def_disp=dvi rootwaitvram=14M' fatload mmc 0 80200000 uImagebootm 80200000

3.1.2. Kernel konfigürasyonu

Kernel CPU, bellek ve I/O gibi donanımın tüm kaynaklarını yöneten yazılımdır. Kernel mimarisiyle birçok platformda çalışan donanımdan bağımsız uygulamalar yazılabilir. Beagleboard-xM ile gelen kernel bu tezde kullanılacak ekran ayarlarında yetersiz kalmaktadır. Bu nedenle kernel kaynak kodu ilgili özellikler açılarak çapraz derlenmiştir.

3.2. DVSDK

DVSDK TI firmasının Linux için dijital video yazılım geliştirme kitidir. Kullanıcıya ARM ve DSP işlemcileri güçlerini birleştirip geliştirme yapmak için her türlü olanağı sağlar. Bu tezde DVSDK 4.01 paket sürümü kullanılmıştır. Paket ile birlikte Arago kök dosya sistemi veDSP/BIOS gerçek zamanlı işletim sistemi de gelmektedir. Ancak bu dosya sisteminde Beagleboard-xM için gereken sürücü paketlerinin tümü bulunmamaktadır. Bu nedenle Beagleboard-xM uyumlu Angström işletim sistemine DVSDK sürücü paketleri eklenerek sistem çalışır hale getirilmiştir.

3.2.1. DVSDK 4.01 kurulumu

DVSDK TI web sitesinden ücretsiz olarak indirilebilir. DVSDK Kütüphanesini Beagleboard destekli düzenlemekiçin resmi olmayan script kullanılır. Beagleboard.org tarafından en son DVSDK 4.01 destekli script yayınlandığı için 4.01 sürümü kullanılmıştır. Bu sürüm sadece Ubuntu10.4 LTS destekli çalıştığı için ana bilgisayarda işletim sistemi Ubuntu 10.4 LTS üzerinde çalışmaktadır.

1. TI web sitesinden TMS320DM3730 DVSDK 4.01 indirilir.
2. Kurulum dosyası yetkilendirilerek çalıştırılır.

- host $chmod 755 dvsdk_dm365-evm_4_00_00_22_setuplinux
- host $./dvsdk_dm365-evm_4_00_00_22_setuplinux

3. Kurulum tamamlandıktan sonra DVSDK psp dosyası altında EVM kartı için bulunan x-loader, u-boot ve kernel dosyaları Beagleboard-xM için olanlarla değiştirilir. Bu amaçla Sourceforge sitesi linkinden RevC-Source.tar dosyası indirilir ve psp dosyası altına açılır.
4. DVSDK'yıBeagleboard uyumlu hale getirmek için script çalıştırılır.

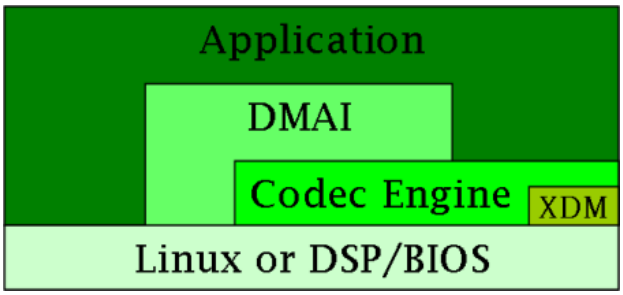
- host $ cd ${HOME}/ti-dvsdk_dm3730-evm_4_01_00_09
- host $ ./setup-dvsdk-4-01-beaglexm-110126.sh

Script çalıştıktan sonra DVSDK'yı derlemek için;

- host $makeclean
- host $makeall

3.2.2. DMAI

Davinci Multimedya Uygulama Arayüzü (DMAI) ARM ve DSP barındıran Davinci platformlarında kolay ve hızlı uygulama yazmak için geliştirilmiş işletim sistemi üzerinde bir katmandır. Linux, DSP/BIOS ve Windows CE gibi işletim sistemleri altında çalışabilir. Bu nedenle işletim sisteminden bağımsız yazılan uygulamalar Davinci platformunu destekleyen donanımlarda çalışır. Şekil 3.2'te görüldüğü üzere DMAI işletim sistemini (OS) veya ses kodlayıcı motorunu (CE) tamamen kapsamaz ancak OS yerine gerektiğinde CE'yi direk olarak kullanabilir.



Şekil 3.2. DMAI blok diyagramı

DMAI'nin önemli avantajlarından bir tanesi de Linux aygıt sürücülerini yönetmek için gerekli kaynakları içermesidir. Örneğin bu çalışmada USB kameradan görüntü almak için DMAI üzerinden Linux V4L2 sürücüleri kullanılmıştır. DMAI DM3730 işlemcisinin Direct Memory Access (DMA) donanım hızlandırıcısını kullanır. DSP Codec Engine, ortak RAM ve Linux kernel sürücüleri arasında hızlı bir köprü görevini görmektedir.

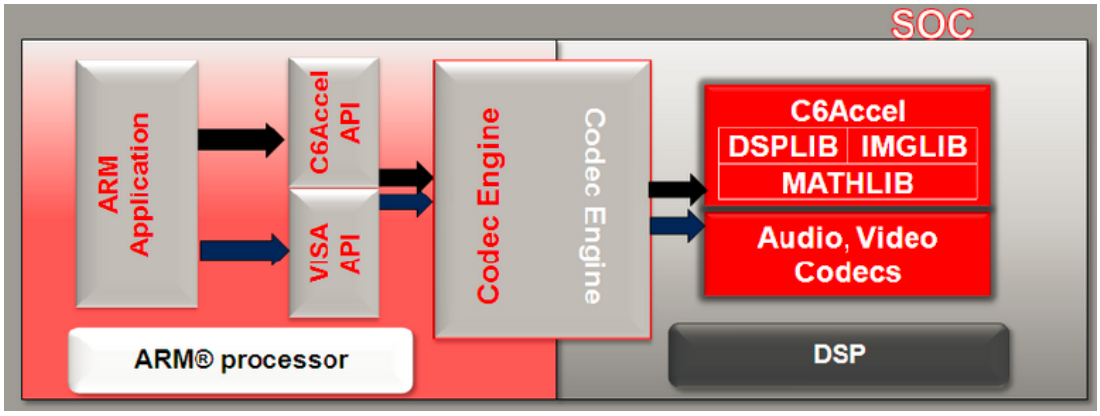
3.2.3. C6Accel

C6Accel ARM+DSPSoC kullanıcılarının ARM tarafından DSP'ye erişmesini sağlayan TI tarafından geliştirilen ücretsiz yazılım geliştirme aracıdır [12]. Özellikle üzerinde Linux koşan ARM tabanlı ve tümleşik DSP bulunduran cihazlar hedeflenmiştir. Sinyal işleme ve dijital görüntü işleme uygulamaları için geliştirilmiş kullanışlı kütüphaneleri kullanıma sunar. Bu kütüphaneler kenar bulma, renk uzayı dönüşümü, FFT ve yoğun matematik işlemleri gibi birçok fonksiyonu içermektedir. C6Accel arayüzü ile kullanılabilen kütüphaneler aşağıda gösterilmiştir.

- C64x+DSPLIB DSP Signal Processing Library version 2.10,
- IMGLIB-2 DSP Image/Video Processing Library Releaseversion 2.0.1,
- IQMathReleaseversion 2.1.3,
- C67xFastRTS Fast Run Time Support Library Release: Version 1.03,
- C674x DSPLIB Digital SignalProcessing Library Release: Version1.2.

Bu tezde özellikle IMGLIB kütüphanesinden yararlanılmıştır. IMGLIB görüntü ve video işleme için son sürüm DSP derleyicileriyle oluşturulmuş ve DSP için yüksek düzeyde optimize edilmiş bir kütüphanedir [14].

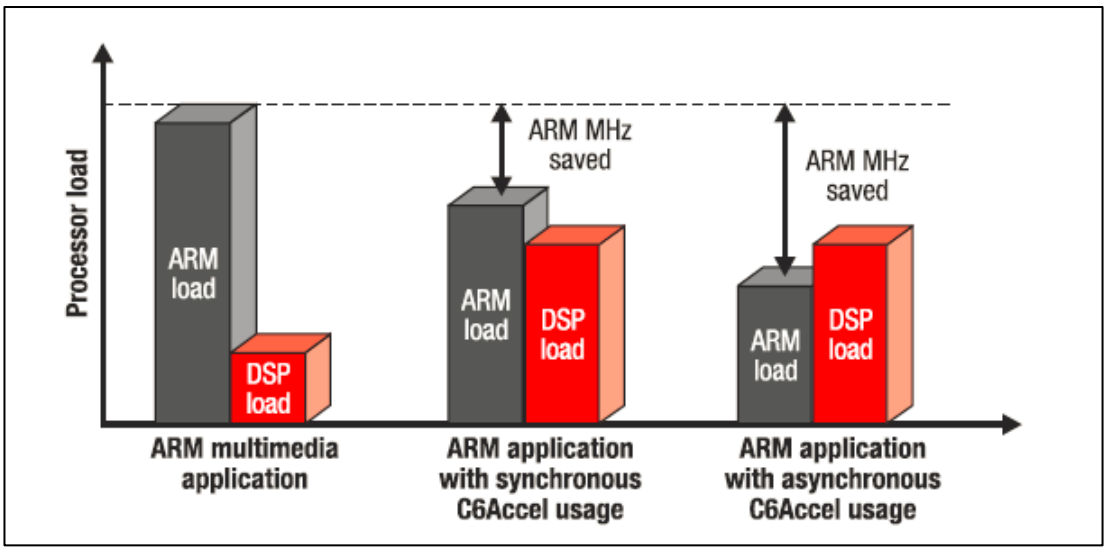
ARM uygulama geliştiricileri DSP'yi genellikle multimedya kodlayıcı için kullanılarakkendilerini limitler. Aslında DSP tarafındaki MHz efektif kullanılarak ilgi uyandıran uygulamalar yazılabilir. C6Accel tarafındaki uygulama C6Accel ile CE (Codec Engine) üzerinden DSP'ye erişir.Şekil 3.3'da görüldüğü üzere ARM uygulaması C6Accel ile DSP tarafındaki DSPLIB, IMGLIB, MATHLIB kütüphaneleri fonksiyonlarını veya AAC, H.264 gibi ses video kodlayıcı-kod çözücü fonksiyonlarını çağırabilir.



Şekil 3.3. C6Accel SoC görünümü

C6Accel yapısının bir diğer önemli avantajı da kullanıcıya kendi oluşturduğu yeni çekirdeği eklemeye imkan tanımasıdır. [15]'te yeni çekirdek ekleme bilgisi detaylı olarak verilmiştir. Bu tezde 1-bit dönüşümü fonksiyonu C6Accel'e bütünleştirilmişve ARM tarafından çağırılarak hem ARM işlemci yükü hafifletilmiş hem de performans artışı sağlanmıştır.

C6Accel senkron ve asenkron çalışma yapısını destekler. Çağırılan fonksiyonların XDAIS algoritma başlıkları ektra yük getirdiğinden fonksiyonların mümkün olduğu kadar birleştirilmesi performans artışı sağlar. Şekil 3.4'te C6Accel'in senkron ve asenkron modları kullanımı durumunda ARM ve DSP işlemci yükü karşılaştırması yapılmıştır. Görüldüğü üzere asenkron mod kullanıldığı durumda ARM işlemci yükü azalır ve diğer uygulamaların çalışmasına olanak sağlar.



Şekil 3.4. C6Accel kullanımı ve işlemci yükü karşılaştırması

3.3. OpenCV

OpenCV gerçek zamanlı uygulamalar hedef alınarak Intel tarafından geliştirilen ve optimize C/C++ ile yazılmış açık kaynaklı Bilgisayarla Görme kütüphanesidir. Linux,Windows ve MacOS gibi işletim sistemlerini desteklemesi ve ücretsiz olması çok tercih edilen bir kütüphane olmasını sağlamıştır.

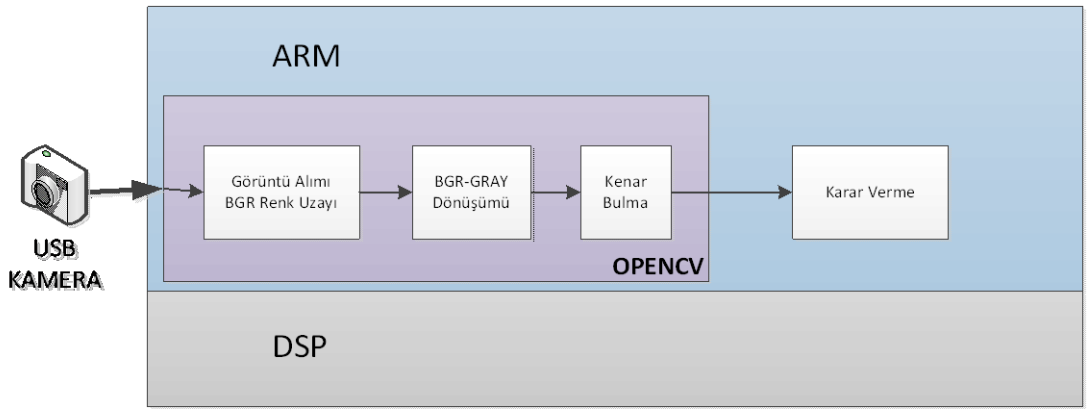
4. UYGULAMA YÖNTEMLERİ

Bu tezde mobil ortamlarda yeterli görüntü kaydetme ve işleme kabiliyetine sahip, düşük güç tüketen TI DM3730 işlemcisini üzerinde barındıran ve kolay uygulama geliştirme ortamı sağlayan Beagleboard-xM gömülü platformu kullanılmaktadır. DM3730 yongası üzerinde işlemci olarak 1-GHz ARM® Cortex™-A8 çekirdeği ve sinyal işleme ünitesi olarak 800-MHz TMS320C64x+TM DSP çekirdeği bulunmaktadır.

Bu çalışmada üç farklı sistem mimarisi kullanılmıştır. Birinci mimaride görüntüyü almak, işlemek ve karar vermek için sadece ARM işlemcisi üzerinde çalışan OpenCV kütüphanesi kullanılmaktadır. İkinci ve üçüncü mimarilerde ise ARM işlemcisi ve DSP birlikte kullanılmaktadır. Aşağıdaki bölümlerde bu üç mimaride kullanılan modüller daha detaylı anlatılmaktadır. Ayrıca bu çözümlerin karşılaştırmalı performans analizi yapılmaktadır.

4.1. ARM Tabanlı OpenCV Kullanımı

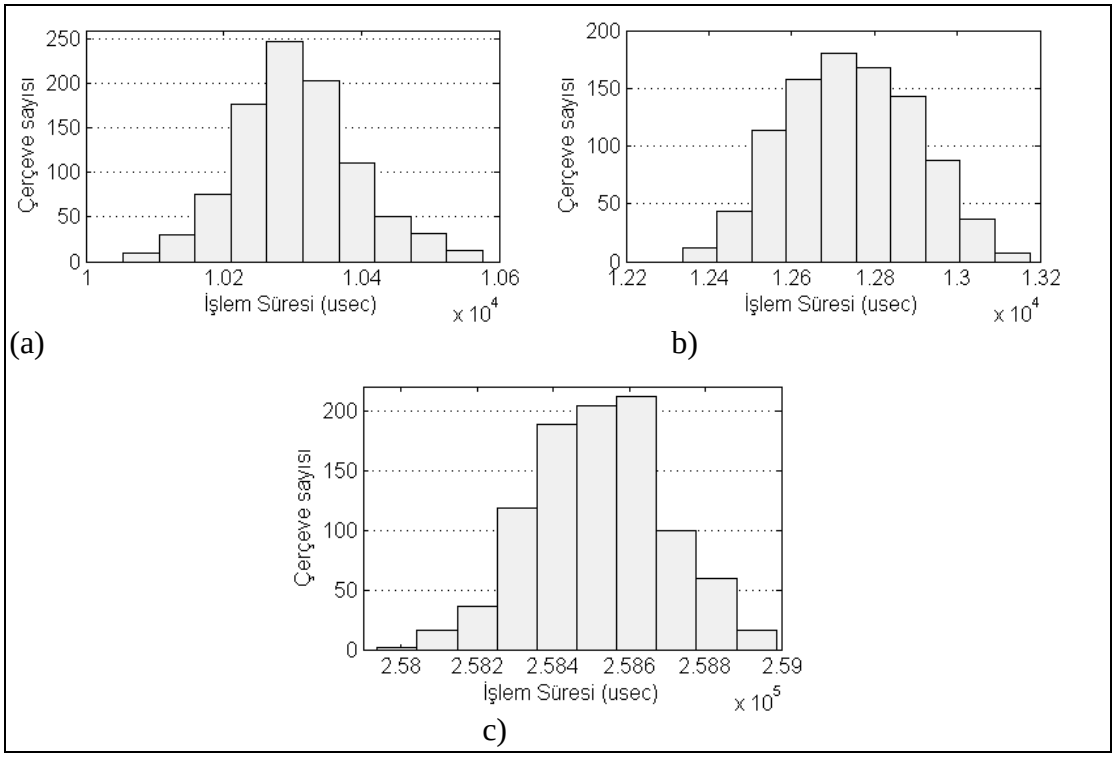
Önerilen ilk sistem mimarisi Şekil 4.1'de gösterilmiştir ve bu sistemde görüntü alma ve işlemleri OpenCV kütüphanesi yardımıyla sadece ARM işlemcisi üzerinde yapılmaktadır. ARM işlemcisi NEON ön işlemcisi desteği sayesinde görüntü işleme hızı artırılmış ve DSP'de işlemeye yakın bir performans vermesi beklenmektedir.



Şekil 4.1. ARM tabanlı sistem mimarisi

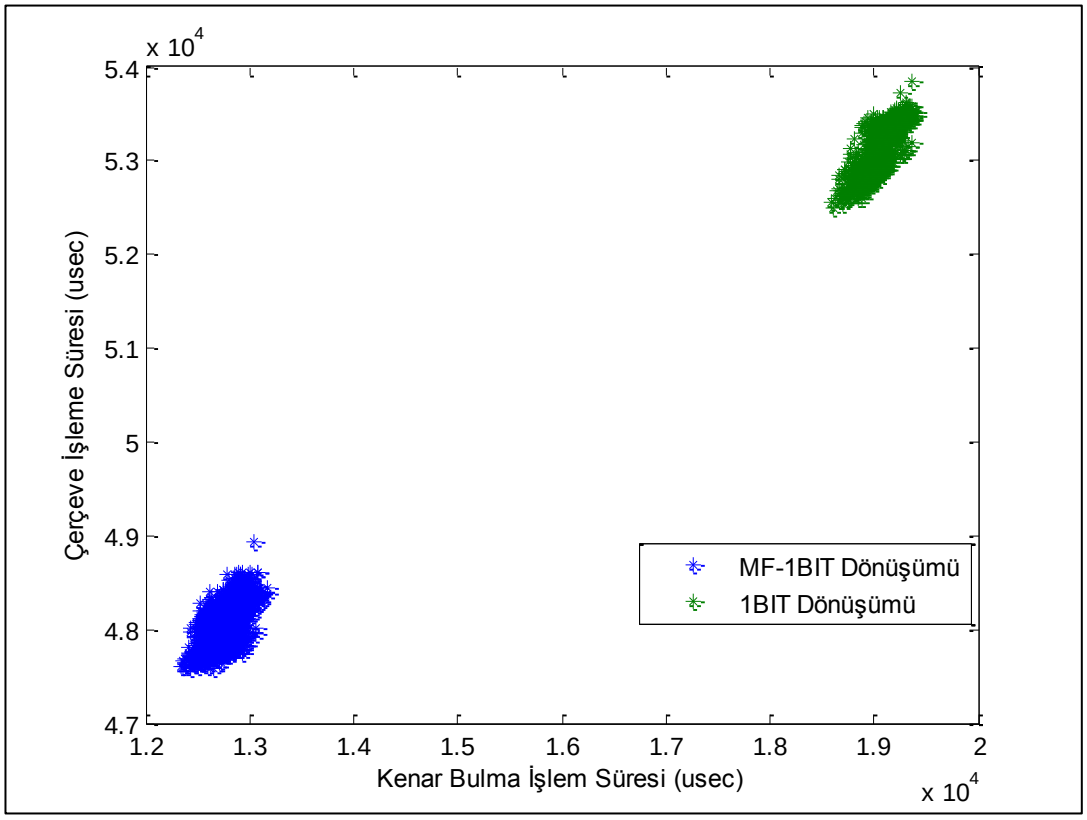
OpenCV ile USB kameradan alınan 640x480 çözünürlüklü görüntü 24bit BGR (mavi-yeşil-kırmızı) formatındadır. Kenar bulma işlemine geçmeden önce görüntüngri formata dönüştürülmesi gerekir. Bu işlem için OpenCV cvtColor() fonksiyonu kullanılmıştır.

Kenar bulma işlemi olarak sobel filtresi, 1-bit dönüşümü ve çarpmasız 1-bit dönüşümü olmak üzere üç farklı yöntem denenmiş ve performans karşılaştırması yapılmıştır. Sobel filtresi işlemi ARM üzerinde OpenCV Kütüphanesi cvSobel() fonksiyonu ile yapılmaktadır. 1-bit dönüşümü ve çarpmasız 1-bit dönüşümleri için ayrı fonksiyonlar tanımlanmıştır. Sobel işlemi bir bit dönüşümüne göre çok daha fazla zaman almaktadır. Performans testi için 640x480 çözünürlüklü USB kameradan 1000 çerçeve alınıp 3 farklı metot ayrı ayrı işlenmektedir. Şekil 4.2'de üç metot için 1000 çerçevenin işlem süresi dağılım figürü ile gösterilmektedir. İşlem süreleri Gauss dağılım sergilemektedir. Görüldüğü üzere sobelfiltesi ortalama 259ms süresi ile açık ara çok daha fazla zaman almaktadır.



Şekil 4.2. ARM üzerinde kenar bulma yöntemleri süre karşılaştırması a) MF-1BIT dönüşümü, b) 1BIT dönüşümü, c) Sobel filtresi

ARM üzerinde çalıştırdığımız 1-bit ve çarpmasız 1-bit dönüşüm süresini yakından incelediğimizde çarpmasız 1-bit dönüşümü ortalama 12,75ms tutarken 1-bit dönüşümü ortalama 19,06ms tutmaktadır.



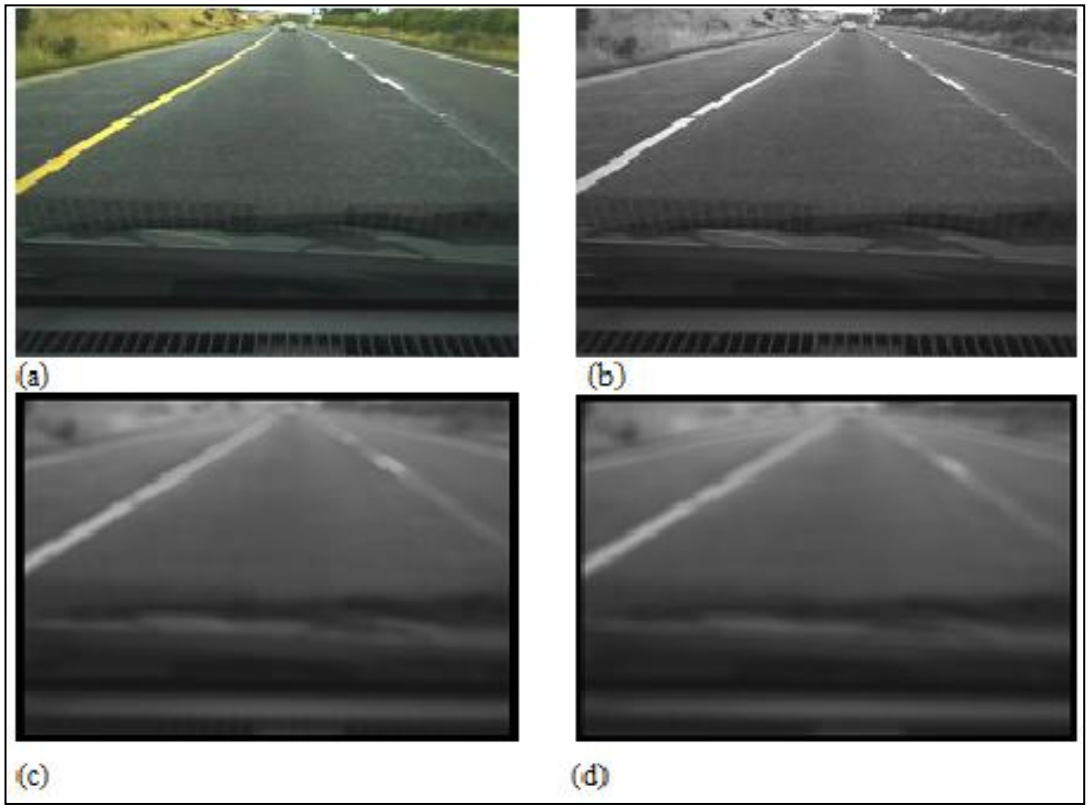
Şekil 4.3. ARM üzerinde MF-1BIT ve 1BIT dönüşümü karşılaştırması

MF-1BIT ve 1BIT dönüşümleri için işlem süresi ve çerçeve işleme süresi karşılaştırması Şekil 4.3'te gösterilmektedir. 1-bit dönüşümü yaklaşık 7ms fazla zaman harcadığından çerçeve işleme süresi ortalama 53,20ms tutmakta ve FPS (saniye işlenen çerçeve) 18,80 olarak hesaplanmaktadır. Çarpmasız 1-bit dönüşümünde ortalama çerçeve işleme süresi 48,12ms tutmakta ve FPS 20,71 olarak hesaplanmaktadır.

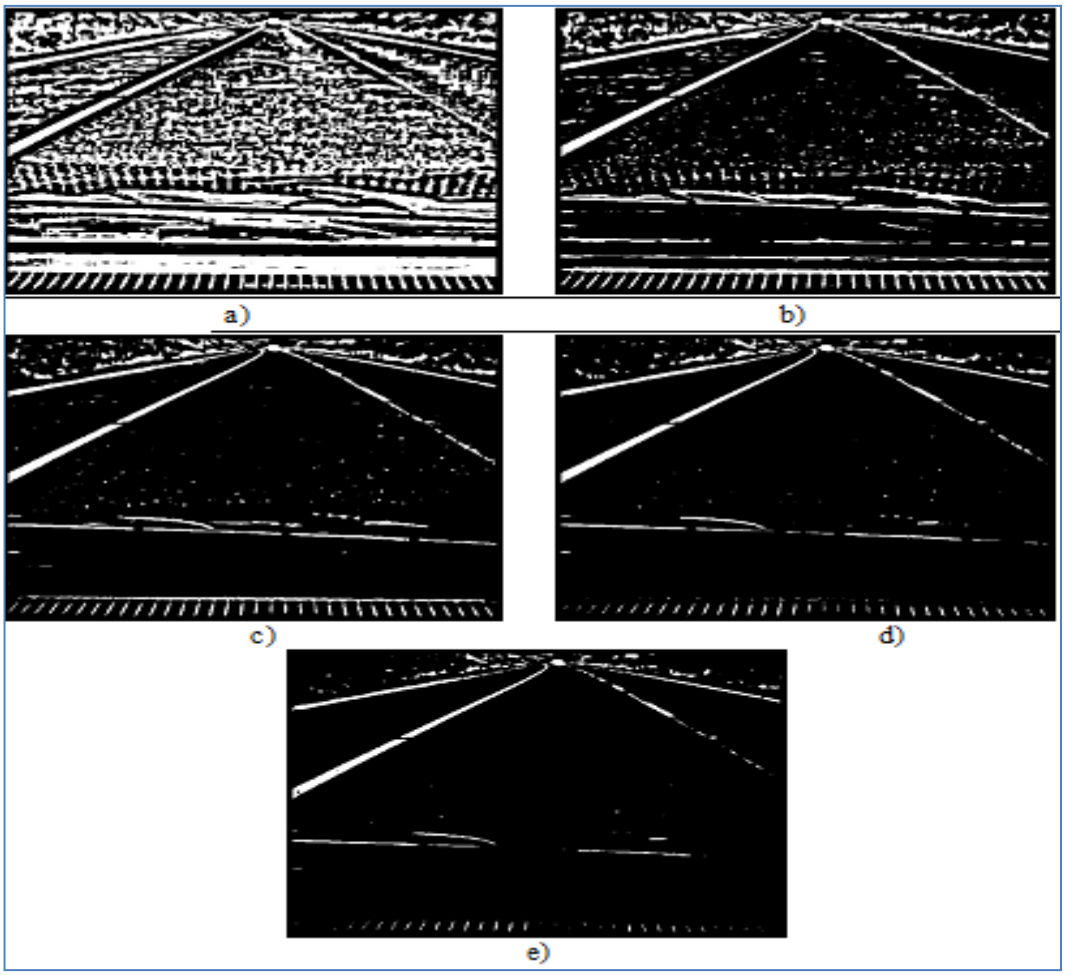
ARM üzerinde üç farklı yöntemin karşılaştırmalı performans sonuçları Tablo 4.1'de verilmiştir. Çarpmasız bir bit dönüşümü sobel filtresine göre 6 kat daha hızlı sonuç vermektedir. Tabloda gösterildiği üzere işlemci yükü her üç yöntem için de %99 olarak ölçülmüştür.

Tablo 4.1. Sistem mimarisi-1 performans sonuçları

	Görüntü Alma	Gri Dönüşümü	Kenar Bulma	Toplam Çerçeve Süresi	FPS	CPU Yükü
MF-1BIT Dönüşümü	23,62ms	11,73ms	12,75ms	48,12ms	20,71	%99
1BIT Dönüşümü	23,28ms	10,86ms	19,06ms	53,20ms	18,80	%99
Sobel Filtresi	17,33ms	9,04ms	259,65ms	286,02ms	3,49	%99



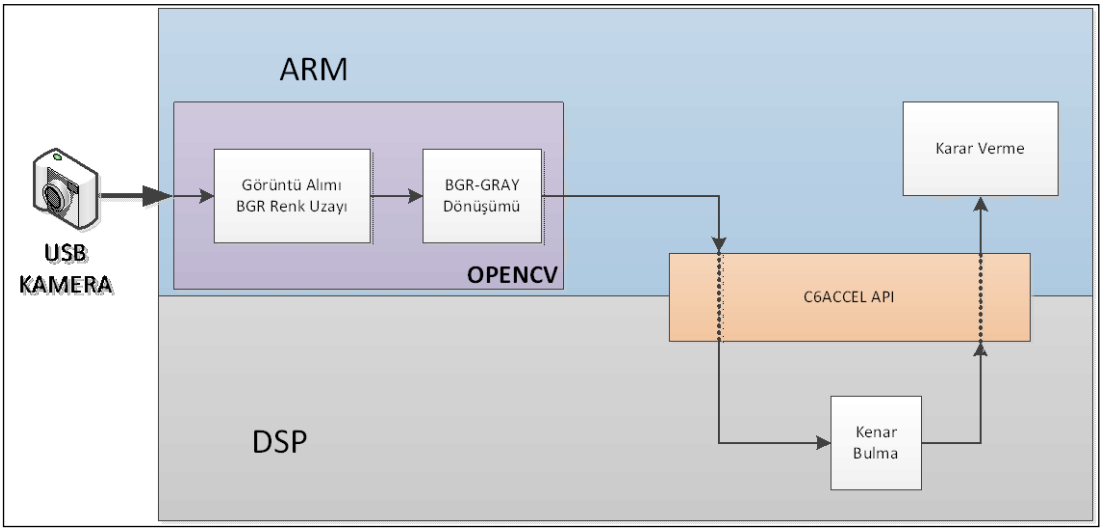
Şekil 4.4. İmge karşılaştırmaları, Orjinal imge(a), Gri (b), 1BT(c), MF-1BT (d)



Şekil 4.5. MF-1BT çeşitli eşik değerleri için çıkış imgeleri a:0, b:5, c:10, d:15, e:20

4.2. ARM Tabanlı OpenCV ve DSP Birlikte Kullanımı

Önerilen ikinci sistem mimarisi Şekil 4.6'da gösterilmiştir ve bu sistemde kenar bulma algoritması ARM işlemci yükünü hafifletmek amacıyla DSP'de gerçeklenmiştir. OpenCV yardımıyla USB kameradan alınan görüntü 24bit BGR formatındadır. Kenar bulma işlemine geçmeden önce görüntü gri formata dönüştürülmesi gerekir. Bu işlem için OpenCV cvtColor() fonksiyonu kullanılmıştır.



Şekil 4.6. ARM ve DSP (C6ACCEL) kullanan sistem mimarisi

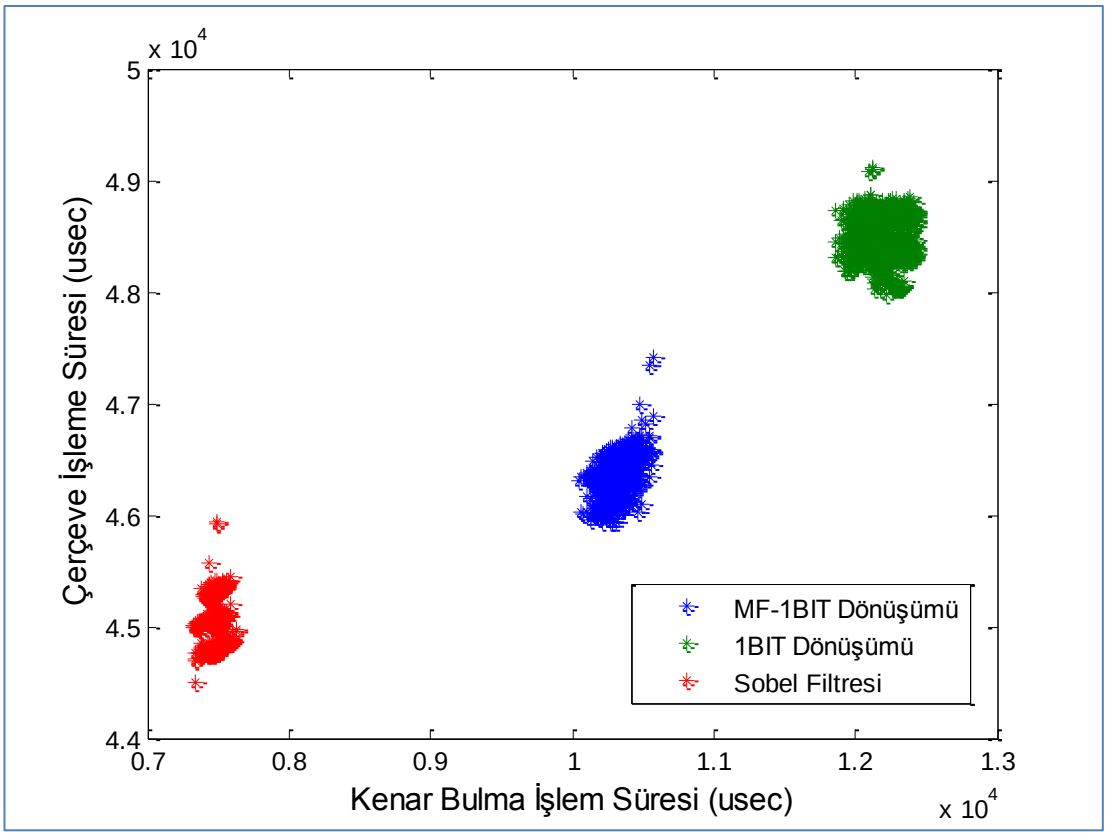
Kenar bulma işlemi olarak sobel filtresi, 1-bit dönüşümü ve çarpmasız 1-bit dönüşümü olmak üzere üç farklı yöntem denenmiş ve performans karşılaştırması yapılmıştır. Sobel filtre işlemini DSP de gerçeklemek için C6Accel arayüzü ile IMGLIB kütüphanesinde bulunan IMG_Sobel_3x3_8() fonksiyonu kullanılmıştır. 1-bit dönüşüm yöntemi için C6Accel'de yeni çekirdek tanımlanmış ve ARM tarafından çağrılmak üzere IMG_1BIT_8() fonksiyonu yaratılmıştır. Bu fonksiyonun parametreleri şunlardır.

- Unsignedchar * in: Gri ton giriş tampon bellek
- Unsignedchar * out: Filtreden geçirilmiş 1-bit çıkış tampon bellek
- UIntRows: Çerçeve genişlik piksel sayısı
- UıntCols: Çerçeve uzunlu piksel sayısı
- Bool mf-1bit: Çarpmasız 1-bit dönüşüm etkin biti

IMG_1BIT_8() fonksiyonu giriş tampon bellek gri tona çevrilmiş çerçeveden oluşmaktadır. Mf-1bit biti ile işlemin 1BT yada MF-1BT olacağı set edilir. Çıkış tampon bellek içeriği fonksiyon çağrılmadan önce sıfırlanmaktadır. Bu sayede 1-bit dönüşümü yapılırken sadece giriş pikselin filtrelenmiş pikselden büyük olduğu durumlarda çıkış tampon belleğine "1" ataması yapılarak işlemci yükü azaltılmıştır.

Sobel filtresi, 1-bit dönüşümü ve çarpmasız 1-bit dönüşüm yöntemlerinin DSP'de gerçekleme sürelerine karşılık ARM tarafında toplam çerçeve işlem süresi Şekil

4.7'de karşılaştırılmıştır. Görüldüğü üzere kenar bulma işlem süresindeki düşüş çerçeve işlem süresinde de düşüşe olanak sağlamıştır.Sobel filtresinin daha hızlı çalışması filtre süzgecinin 1BT'de kullanılan 17x17'lik yerine daha küçük 3x3 olması olarak açıklanabilir. MF-1BT'in kayan noktalı işlem olmadığı için 1-BT'ye göre daha hızlı çalıştığı gözlemlenmektedir.



Şekil 4.7. DSP'de C6Accel kullanarak kenar bulma süreleri karşılaştırması

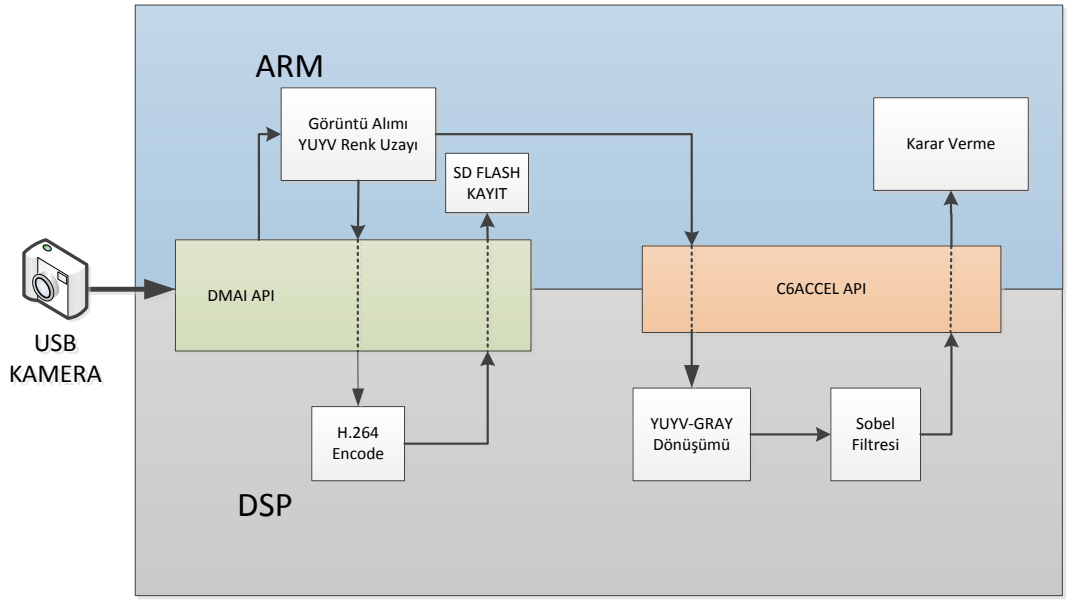
Sistem mimarisi-2'nin performans sonuçları Tablo 4.2'de verilmiştir. 1-bit dönüşümü yaklaşık çerçeve işleme süresi ortalama 48.50ms tutmakta ve saniye işlenen çerçeve (FPS) 20,61 olarak hesaplanmaktadır. Çarpmasız 1-bit dönüşümünde ortalama çerçeve işleme süresi 46,43ms tutmakta ve FPS 21,53 olarak hesaplanmaktadır. En iyi performansı 46,03ms çerçeve işleme süresiyle sobel filtresi göstermiştir. İşlemci yükü (CPU yükü) sistem mimarisi-1 ile karşılaştırıldığında % 99 mertebesinden % 75 seviyelerine inmiştir.

Tablo 4.2. Sistem mimarisi-2 performans sonuçları

	Görüntü Alma	Gri Dönüşümü	Kenar Bulma	Toplam Çerçeve	FPS	CPU Yükü
MF-1BIT Dönüşümü	24,13ms	11,98ms	10,31ms	46,43ms	21,53	%75
1BIT Dönüşümü	24,26ms	12,05ms	12,16ms	48,50ms	20,61	%79
Sobel Filtresi	25,73ms	12,82ms	7,48ms	46,03ms	21,72	%81

4.3. ARM ve DSP Birlikte Kullanımı

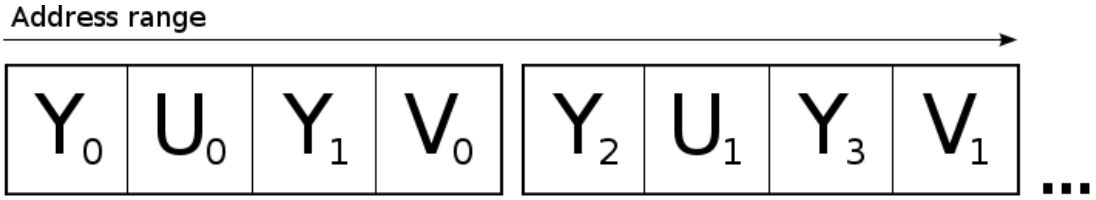
Önerilen üçüncü sistem mimarisinde DVSDK geliştirme kütüphanesinin DMAI arayüzünün sağladığı olanaklardan yararlanarak performans hızında artış ve işlemci yükünde azalma hedeflenmektedir. Ayrıca DMAI arayüzünün kolayca kullanıma sunduğu multimedya kodlayıcı H.264 sıkıştırma işlemi sisteme eklenerek sistemin olanakları analiz edilmektedir. Sistem mimarisi Şekil 4.8'de gösterilmektedir.



Şekil 4.8. ARM ve DSP (DMAI ve C6ACCEL) kullanan sistem mimarisi

Bu sistemin en önemli bileşeni DMAI arayüzüdür. DMAI DM3730 işlemcisinin Direct Memory Access (DMA) donanım hızlandırıcısını kullanır. Linux çekirdeğini yönetmek ve düzenlemek için gerekli kaynak kodlarını içerir. DM3730 çekirdeği V4L2 sürücülerini içerir, dolayısıyla YUV formatındaki çerçeveleri kameradan yakalayıp ekranda gösterebilmektedir.

YUYV piksel formatı YUV renk uzayının bir alt kümesidir. YUV renk uzayı renkli görüntüyü sıkıştırır ve renk bileşenleri için daha düşük bir bant genişliği sağlar. Bu sayede direk RGB gösterimine göre iletim hatalarına ve sıkıştırma yapaylıklarına karşı daha etkili maskeleme yapar. RGB formatında her piksel üç bayt ile ifade edilirken YUYV formatında her iki piksel dört bayt ile tanımlanır. Birinci ve üçüncü (Y) baytlar Luma (parlaklık) bileşenleridir. İkinci ve dördüncü baytlar (UV) iki renk bileşenini temsil eder.



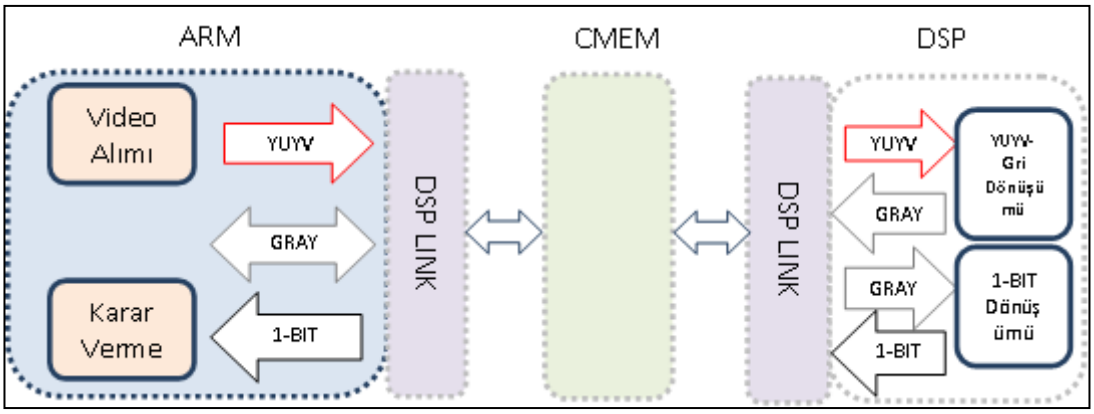
Şekil 4.9. YUYV piksel format

YUV format gösterimi paketli format olduğundan insanın gözünün algılayabileceği bir görüntü değildir. Bu nedenle RGB dönüşümü yapılarak ekrana basılması gerekmektedir. Aşağıda denklem (4.1)'de YUV-RGB dönüşüm denklem matrisi verilmiştir.

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,14713 & -0,28886 & 0,436 \\ 0,615 & -0,51499 & -0,10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.1}$$

V4L2 Linuxekran sürücüsüne gönderilen YUYV paket formatlı çerçeve otomatik olarak RGB'ye çevrilmekte ve ekrana basılmaktadır. DM3730 çekirdeğinde YUYV formatından başka RGB565 formatı da desteklenmektedir.

Alınan YUYV çerçevesinin kenar bulma işlemine girebilmesi için önce gri tona çevrilmesi gerekmektedir. DSP üzerinde 1-bit dönüşüm blok diyagramı Şekil 4.10'da gösterilmektedir.



Şekil 4.10.DSP Üzerinde 1-BIT dönüşümü blok diyagramı

DSP'de işlenecek verilerin CMEM ortak bellek alanına yazılması gerekir. Diyagramdan görüldüğü üzere kameradan alınan YUYV çerçeveleri önce CMEM modülünde tanımlı olan ortak bellek alanına yazılır. Daha sonra ARM üzerinden DSPLink vasıtasıyla DSP'ye işlem komutu gönderilir. DSP ortak bellek alanından veriyi alır işler ve tekrar ortak belleğe yazar.

Bu sistemde alınan YUYV formatındaki çerçeveleri gri tona çevirmek için C6Accel arayüzünde tanımlı C6accel_IMG_yc_demux_8() fonksiyonu kullanılmıştır. Fonksiyonun aldığı parametreler şunlardır:

- hC6accel: C6Accel handle
- n :Number of lumapixels
- yc: Interleavedluma/chroma
- y : Lumaplane (8-bit)
- cr: Cr chromaplane (8-bit)
- cb: Cbchromaplane (8-bit)
- Return value: Errorcodes

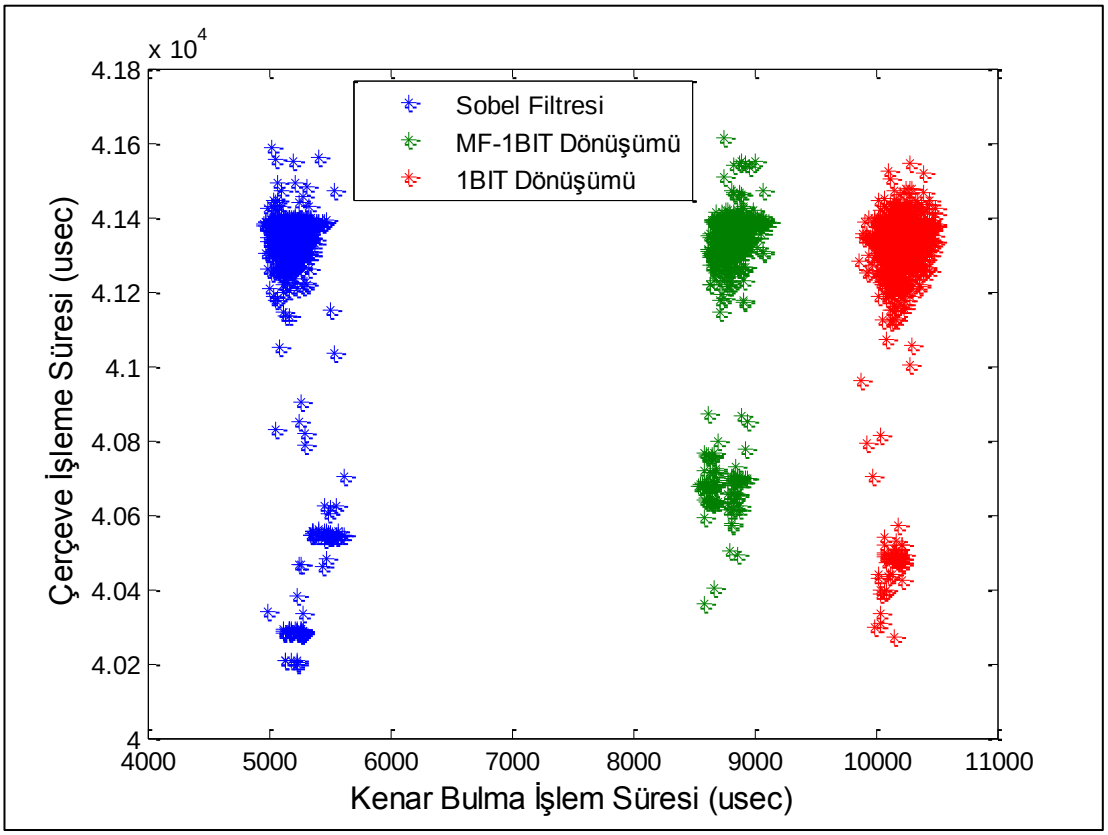
YUYV formatında her piksel 2 bayt ile temsil edildiğinden giriş buffer uzunluğu şöyle hesaplanır,

Giriş tampon = 640 x 480 x 2 = 614400 (4.2)

C6accel_IMG_yc_demux_8() fonksiyonu çıktısında gri tonda veri her piksel 1 bayt ile ifade edilir. O halde çıkış tampon bellek gri ton olacağı için uzunluğu şöyle hesaplanır,

Çıkış tampon = 640 x 480 = 307200 (4.3)

Sobel filtresi, 1-bit dönüşümü ve çarpmasız 1-bit dönüşüm yöntemlerinin DSP'de DMAI arayüzü kullanılarak gerçekleme sürelerine karşılık ARM tarafında toplam çerçeve işlem süresi Şekil 4.11'de karşılaştırılmıştır. Görüldüğü üzere kenar bulma işlem süresindeki düşüş çerçeve işlem süresinde düşüşe sebep olmamıştır. Bunun nedeni kullandığımız A4Tech 640x480 çözünürlüklü kameranın 24 FPS ile sınırlı olarak çalışmasıdır. Sobel filtresinin daha hızlı çalışması filtre süzgecinin 1BT'de kullanılan 17x17'lik yerine daha küçük 3x3 olması olarak açıklanabilir. MF-1BT'in kayan noktalı işlem olmadığı için 1-BT'ye göre daha hızlı çalıştığı gözlemlenmektedir.



Şekil 4.11. DSP DMAI kullanarak kenar bulma süreleri karşılaştırması

Sistem mimarisi-3'ün performans sonuçları Tablo 4.3'te verilmiştir. 1-bit dönüşümü yaklaşık çerçeve işleme süresi ortalama 41.23ms tutmakta ve saniye işlenen çerçeve (FPS) 24,25 olarak hesaplanmaktadır. Çarpmasız 1-bit dönüşümünde ortalama çerçeve işleme süresi 41,20ms tutmakta ve FPS 24,25 olarak hesaplanmaktadır. Sobel filtresinde ise ortalama çerçeve işleme süresi 41,19ms tutmakta ve FPS 24,27

olarak hesaplanmaktadır. İşlemci yükü (CPU yükü) sistem mimarisi-1 ve sistem mimarisi-2 ile karşılaştırıldığında ortalama %53 hafiflemiştir. Üç yöntemin de yaklaşık % 34 işlemci yükü kullanması ARM ve DSP tabanlı sistemlerde görüntü işlemenin DSP üzerinde yapılmasının avantajını göstermektedir.

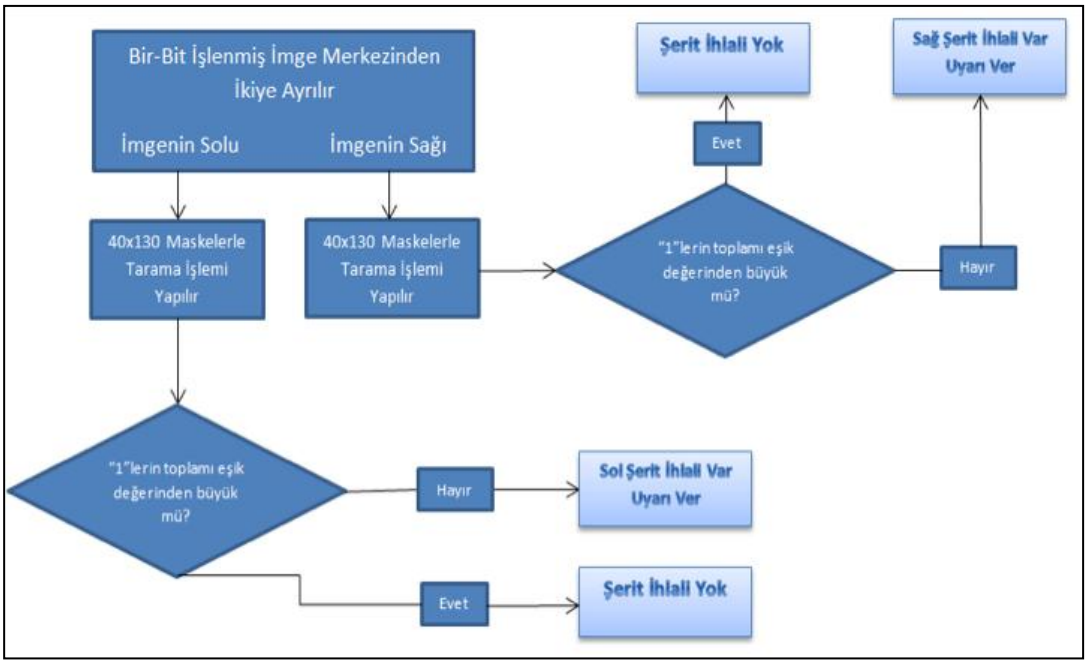
Sistem mimarisi-3 önceki iki sistemler karşılaştırıldığında hem FPS'de artış hem de CPU yükünde azalma sağlamıştır.

Tablo 4.3. Sistem mimarisi-3 performans sonuçları

	Görüntü Alma	Gri Dönüşümü	Kenar Bulma	Toplam Çerçeve	FPS	CPU Yükü
MF-1BIT Dönüşümü	19,9ms	9,5ms	8,8ms	41,2ms	24,25	%34
1BIT Dönüşümü	18,3ms	9,6ms	10,19ms	41,23ms	24,25	%35
Sobel Filtresi	23,46ms	9,53ms	5,18ms	41,19ms	24,27	%36

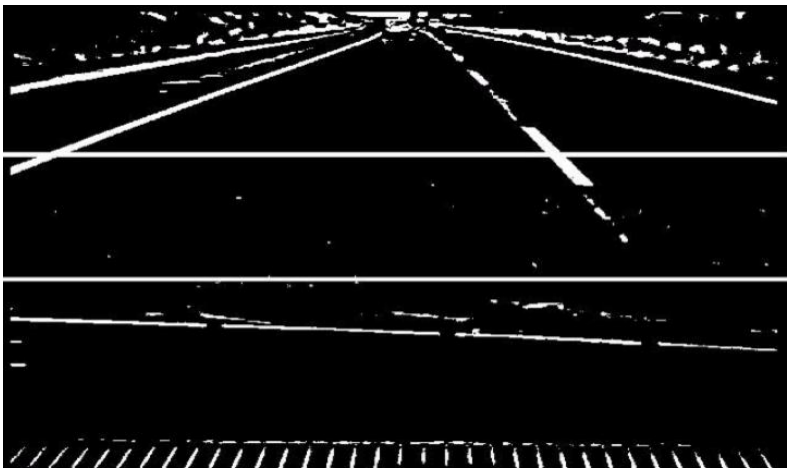
4.4. Şerit Tespit Karar Algoritması

Yol üzerinde tespit edilmek istenen çizginin belirli bir yerde olması gerektiği bilinmektedir. Bu nedenle yol çizgisinin bulunması için tüm görüntünün taranması yerine sınırları belirlenmiş özel bir bölgede ROI (ilgilenen bölge)aranması tercih edilmektedir [14]. Bu sayede işlemci yükü azalıp şerit tespit algoritması çalışma zamanı da kısalmaktadır. Şekil 4.14'te ROI bölgesi gösterilmiştir.



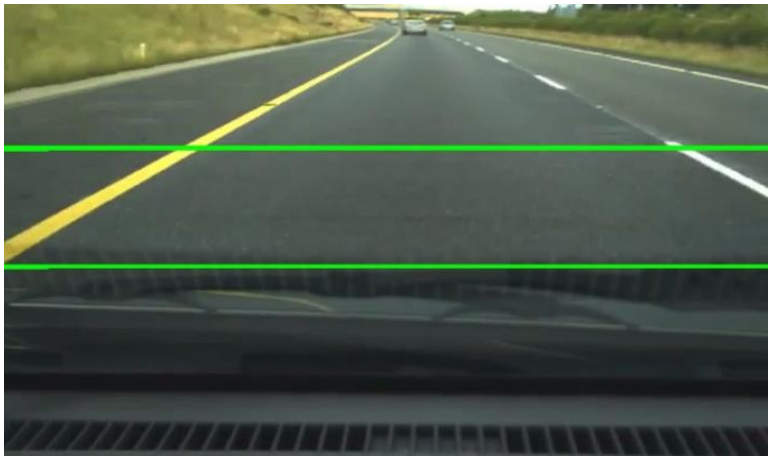
Şekil 4.12. Şerit tespit karar algoritması diyagramı

Şerit tespit algoritmasının diyagramı Şekil 4.12'de verilmiştir. Buna göre 1BT işleminden geçmiş her çerçeve sırasıyla işlenmektedir. İşlenecek imge "imgenin sağı" ve "imgenin solu" olmak üzere ortadan iki bölgeye ayrılır.



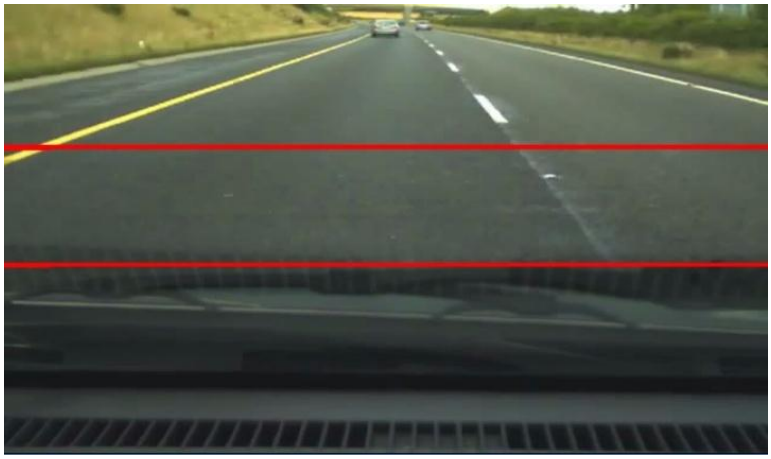
Şekil 4.13. 1BT işlenmiş ve ROI belirlenmiş imge

Her bölgede ROI alanında 40x130 piksel ölçülü maskelerle tarama işlemi yapılır. Tarama sonucunda "1" bulunan piksellerin toplamının en çok olduğu bölgenin piksel koordinatı bulunur. Bu koordinatın yatay eksende orta noktaya uzaklığı belirlenen eşik değerinden fazla ise şerit ihlali uyarısı verilmektedir. Şekil 4.13'te 1BT dönüşümü yapılmış ve ROI bölgesi belirlenmiş çerçeve gösterilmektedir.



Şekil 4.14. Araç şerit içinde durumu

Araç şerit içindeyken ROI bölgesi yeşil boyalı gösterilmektedir. Şekil 4.14'te aracın şerit ihlali olmayan durumu gösterilmektedir.



Şekil 4.15. Araç sağ bölgeden şerit ihlali durumu

Araç şerit dışına çıkmaya yakın duruma geldiğinde ROI bölgesi kırmızıya boyanarak sürücü uyarılmaktadır. Araç şerit içine girdiğinde ROI bölgesi tekrar yeşil olmaktadır. Örneğin Şekil 4.15'te sağ şeritten ayrılan aracın uyarılma durumu gösterilmektedir.

5. SONUÇLAR VE ÖNERİLER

Bu tezde şerit tespit sisteminin gerçek zamanlı gömülü sistem olan BeagleBoard-xM platformu üzerinde gerçeklenmesi ve üç farklı sistem mimarisi için performans analizi yapılmıştır. Birinci mimaride görüntüyü almak, işlemek ve anlamlandırmak için sadece ARM işlemcisi üzerinde çalışan OpenCV kütüphanesi kullanılırken diğer iki mimaride ise ARM işlemcisi ve DSP birlikte kullanılmaktadır. Şerit tespit sisteminin önemli bileşenlerinden kenar bulma algoritması olarak literatürde yaygın olarak kullanılan sobel filtresinin yanı sıra 1-bit dönüşümü ve çarpmasız 1-bit dönüşümü olmak üzere üç farklı yöntem uygulandı. Gerçekleme için sadece ARM işlemcisi kullanıldığında işletim sisteminin sağladığı avantajlardan yararlanılırken gerçek zamanlı uygulamanın gerektirdiği zaman ölçütleri açısından düşük performans gözlemlendi. DSP çekirdeğinin kullanıldığı diğer iki mimaride ise görüntü çerçeveleri daha hızlı bir şekilde işlenirken ARM üzerindeki işlemci yükü %100'den %34 seviyelerine düşerek azalmıştır.

İleri aşamada USB kamera yerine paralel port ara yüzüne sahip bir kamera kullanıldığında işlemci yükü %10 seviyelerine düşeceği öngörülmektedir. Bunun nedeni USB kameradan alınan çerçeveler Linux üzerinden CMEM ortak belleğine kopyalanırken işlemci kullanılmaktadır. Paralel port kamerada ise alınan çerçeveler DMA yardımıyla direk CMEM ortak belleğine yazılmaktadır. Aynı zamanda 640x480'lik bir çerçevenin ortalama 4ms olan kopyalama süresi ortadan kalktığı için işlem süresinden de kazanç sağlanacaktır.

KAYNAKLAR

[1] T.C. ULAŞTIRMA, DENİZCİLİK VE HABERLEŞME BAKANLIĞI KARAYOLLARI GENEL MÜDÜRLÜĞÜ Trafik Kazaları Özeti 2012, http://www.kgm.gov.tr/SiteCollectionDocuments/KGMdocuments/Trafik/TRAFIKKAZA%20OZETI2012.pdf (Ziyaret tarihi: 10 Mayıs 2014).

[2] Google Driverless Car, http://en.wikipedia.org/wiki/Google_driverless_car (Ziyaret tarihi: 10 Mayıs 2014).

[3] Maini R., Aggarwal H., Study and Comparison of Various Image Edge Detection Techniques, *International Journal of Image Processing*, 2009, **3**(1), 1-11

[4] Natarajan B., Bhaskaran V., Konstantinides K., Low-complexity block-based motion estimation via one-bit transforms, *IEEE Trans.CircuitSyst. Video Technol.*, 1997, **7**(5), 702-706.

[5] Ertürk S., Multiplication-FreeOne-Bit Transform for Low-Complexity Block Based Motion Estimation, *IEEE, Signal Processing Letters*, 2000, **14**(2), 109-112.

[6] BeagleBoard-xMRev C2 System Reference Manual Revision0.0 October 11, 2012 (Ziyaret tarihi: 12 Mayıs 2014).

[7] BeagleBoard-xM, http://www.watterott.com/en/BeagleBoard-xM (Ziyaret tarihi: 12 Mayıs 2014).

[8] DM3730, DM3725 Digital Media Processor, http://www.ti.com/lit/ds/symlink/dm3725.pdf (Ziyaret tarihi: 16 Haziran 2014).

[9] Carlson B.,Giolma B.,SmartReflex powerand performance management technologies: reduced power consumption, optimized performance, http://www.ti.com/lit/wp/swpy015a/swpy015a.pdf (Ziyaret tarihi: 27 Mayıs 2014).

[10] ARM Cortex-A8 Processor, http://www.arm.com/products/processors/cortex-a/cortex-a8.php (Ziyaret tarihi: 27 Mayıs 2014).

[11] NEON, http://www.arm.com/products/processors/technologies/neon.php (Ziyaret tarihi: 3 Haziran 2014).

[12] Texas Instruments. CMEM Overview, http://processors.wiki.ti.com/index.php/CMEM_Overview (Ziyaret tarihi: 3 Haziran 2014).

[13] C6ACCEL, http://processors.wiki.ti.com/index.php/C6EZAccel_FAQ (Ziyaret tarihi: 3 Haziran 2014).

[14] IMGLIB, http://processors.wiki.ti.com/index.php/IMGLIB (Ziyaret tarihi: 11 Haziran 2014).

[15] C6Accel Advanced User Guide, http://processors.wiki.ti.com/index.php/C6Accel_Advanced_Users_Guide#Adding_a_new_kernel.2Flibrary_to_C6Accel (Ziyaret tarihi: 15 Haziran 2014).

[16] Lin Q., Han Y., Hahn H., Real-time lane departure detection based on extended edge-linking algorithm, *Second International Conference on Computer Research and Development*, Kuala Lumpur, Malezya, 7-10 Mayıs 2010.

KİŞİSEL YAYIN ve ESERLER

[1] Ilgar M., **Küyük R. T.**, Ertürk S., Şeritten Ayrılma Uyarı Sisteminin ARM ve DSP Üzerinde Karşılaştırmalı Uygulaması, *Gömülü Sistemler ve Uygulamaları Sempozyumu*, İstanbul, Türkiye, 4-5 Aralık 2014.

ÖZGEÇMİŞ

1985 yılında Balıkesir'de doğdu. İlk ve orta öğrenimini Balıkesir Sırrı Yırcalı Anadolu Lisesi'nde tamamladı. 2003 yılında girdiği İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi Kontrol Mühendisliği Bölümü'nden 2008 yılında mezun oldu. Aynı yıl TÜBİTAK Gebze Yerleşkesi TEKGEB'te (Teknoloji Geliştirme Bölgesi) bulunan Teknoset firmasında Kablosuz Sensör Ağları Uygulamaları konularında Ar-Ge çalışmaları yapmak üzere işe başladı ve 4,5 yıl boyunca devam etti.

2012 yılında TÜBİTAK BİLGEM UEAKE'de (Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü) Araştırmacı olarak göreve başladı. Halen bu görevine devam etmektedir.