



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**ELEKTRONİK SİSTEMLERLE DİŞ RENGİNİN
TESPİT EDİLMESİ**

Aytaç ALTAN

Biyomedikal Mühendisliği Anabilim Dalı

Danışman

Prof.Dr. Aydın AKAN

Ocak, 2011


İSTANBUL

Bu çalışma 13/01/2011 tarihinde aşağıdaki jüri tarafından Biyomedikal Mühendisliği Anabilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi


Prof. Dr. Aydın AKAN (Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi


Prof. Dr. Mukden UĞUR
İstanbul Üniversitesi
Mühendislik Fakültesi


Yrd. Doç. Dr. Mana SEZDİ
İstanbul Üniversitesi
Teknik Bilimler M. Y. O.


Yrd. Doç. Dr. Niyazi ODABAŞIOĞLU
İstanbul Üniversitesi
Mühendislik Fakültesi


Yrd. Doç. Dr. Niyazi KILIÇ
İstanbul Üniversitesi
Mühendislik Fakültesi

İÇİNDEKİLER

İÇİNDEKİLER	ii
ŞEKİL LİSTESİ	iv
TABLO LİSTESİ	vi
ÖZET	vii
SUMMARY	viii
1. GİRİŞ	1
2. GENEL KISIMLAR	3
2.1. DIŞ HEKİMLİĞİNDE RENK SEÇİMİ	3
2.1.1. Metamerizm	4
2.1.2. Diş Hekimliğinde Kullanılan Renk Sistemleri	4
2.1.3. Dişlerin Translusens ve Floresans Özellikleri	6
2.1.4. Dişlerin Renk Özellikleri	6
2.1.5. Renk Skalaları	7
2.1.6. Renk Seçimi	9
2.1.7. Renk Seçiminde Genel Prensipler	11
2.1.8. Görsel Renk Seçiminin Dezavantajları	12
2.1.9. Diş Rengi Seçiminde Teknolojik Gelişmeler	13
2.1.10. Piyasadaki Popüler Renk Seçim Cihazları	15
2.2. SAYISAL GÖRÜNTÜ İŞLEME	18
2.2.1. Sayısal Görüntü İşleme Sisteminin Temel Öğeleri	18
2.2.2. Sayısal Görüntünün Bazı Temel Kavramları	21
2.2.2.1. Siyah - Beyaz Görüntü	21
2.2.2.2. Sayısal Görüntü	22
2.2.2.3. Renkli Görüntü	23
2.2.2.4. Görüntü Dosyaları	23
2.2.2.5. Piksel	23

2.2.2.6. Çözünürlük	23
2.2.2.7. Sıkıştırma	24
2.2.3. Görüntü Pekiştirme	24
2.2.4. Filtreleme	25
2.2.5. Medyan Filtresi	26
2.3. RENK UZAYLARI	26
2.3.1. RGB renk uzayı	27
2.3.2. HSI, HLS ve HSV renk uzayları	28
2.3.3. RGB Renk Uzayından HSV Renk Uzayına Dönüşüm	30
3. MALZEME VE YÖNTEM	31
3.1. YAZILIM GELİŞTİRME ORTAMININ KURULUMU	31
3.1.1. Microsoft Visual Studio 2005	31
3.1.2. OpenCV	32
3.1.3. Microsoft Visual Studio 2005 üzerine OpenCV kurulumu	33
3.2. YÖNTEM	34
3.2.1. Gerçek Zamanlı Görüntü İşleme Yöntemi	34
3.2.1.1. Kalibrasyon	35
3.2.1.2. Renk Ölçümü	36
3.2.2. Durdurulmuş Görüntüyü İşleme Yöntemi	38
3.2.2.1. Kalibrasyon	38
3.2.2.2. Renk Ölçümü	39
4. BULGULAR	42
4.1. Gerçek Zamanlı Görüntü İşleme Yöntemindeki Bulgular	42
4.1. Durdurulmuş Görüntüyü İşleme Yöntemindeki Bulgular	44
5. TARTIŞMA VE SONUÇ	47
KAYNAKLAR	48
EKLER	50
ÖZGEÇMİŞ	69

ŞEKİL LİSTESİ

Şekil 2.1	: Nesnelerin görülebilmesi için ışık yaymaları veya yansıtmaları	3
Şekil 2.2	: Farklı kaynaklar altında farklı algılanan renkler	4
Şekil 2.3	: Munsell renk sistemi	5
Şekil 2.4	: Munsell renk sisteminin renk tonlarına göre gösterimi	6
Şekil 2.5	: Vita Classical renk skalası	7
Şekil 2.6	: Ivoclar Chromascop renk skalası	8
Şekil 2.7	: Vita 3D Master renk skalası	9
Şekil 2.8	: Renk skalası kullanılarak hasta dişine uygun renk seçimi	11
Şekil 2.9	: Shofu ShadeEye cihazı	15
Şekil 2.10	: Cynovad Shadescan cihazı	16
Şekil 2.11	: MHT SpectroShade cihazı	17
Şekil 2.12	: Vita EasyShade cihazı	17
Şekil 2.13	: X-Rite ShadeVision cihazı	18
Şekil 2.14	: Sayısal görüntü işleme sisteminin temel öğeleri	20
Şekil 2.15	: Sayısal görüntü ve piksel	24
Şekil 2.16	: Filtreleme işleminin kabaca gösterimi	26
Şekil 2.17	: 3x3'lük medyan filtresi	26
Şekil 2.18	: RGB renk uzayı	28
Şekil 2.19	: HLS renk uzayı	28
Şekil 2.20	: Altıgen piramit şeklindeki HSV renk uzayı	29
Şekil 2.21	: Çift altıgen piramit şeklinde HSI renk modeli	29
Şekil 2.22	: RGB-HSV dönüşüm formülleri	30
Şekil 3.1	: Microsoft Visual Studio 2005	31
Şekil 3.2	: OpenCV	32
Şekil 3.3	: OpenCV'nin Visual Studio üzerine kurulumu	33
Şekil 3.4	: USB ağız içi kamera	35
Şekil 3.5	: Kameranın lens ve ledlerden oluşan uç kısmı	35
Şekil 3.6	: Gerçek zamanlı görüntü işleme yöntemi için tasarlanan kalibrasyon yazılımının algoritması	36
Şekil 3.7	: Gerçek zamanlı görüntü işleme yöntemi için tasarlanan renk ölçümü yazılımının algoritması	37
Şekil 3.8	: Logitech Webcam Pro 9000	38
Şekil 3.9	: Durdurulan görüntüyü işleme yöntemi için tasarlanmış kalibrasyon algoritması	40
Şekil 3.10	: Durdurulan görüntüyü işleme yöntemi için tasarlanmış renk ölçümü algoritması	41
Şekil 4.1	: Gerçek zamanlı görüntü işleme yöntemindeki kalibrasyon aşaması	42

Şekil 4.2	: Gerçek zamanlı görüntü işleme yöntemindeki renk ölçümü aşaması	43
Şekil 4.3	: Durdurulmuş görüntüyü işleme yöntemindeki kalibrasyon aşaması (1)	45
Şekil 4.4	: Durdurulmuş görüntüyü işleme yöntemindeki kalibrasyon aşaması (2)	45

TABLO LİSTESİ

Tablo 2.1	: Küp köşegenlerinde elde edilen renk tonları	27
Tablo 4.1	: Gerçek zamanlı görüntü işleme yönteminde elde edilen kalibrasyon sonuçları	43
Tablo 4.2	: Gerçek zamanlı görüntü işleme yönteminde elde edilen renk ölçümü sonuçları	44
Tablo 4.3	: Durdurulmuş görüntüyü işleme yönteminde elde edilen kalibrasyon sonuçları	46
Tablo 4.4	: Durdurulmuş görüntüyü işleme yönteminde elde edilen renk ölçümü sonuçları	46

ÖZET

ELEKTRONİK SİSTEMLERLE DİŞ RENGİNİN TESPİT EDİLMESİ

Estetik diş hekimliğinde renk belirlenmesi görsel ve aletli olmak üzere iki şekilde yapılmaktadır. Hekimin subjektif görüşüne dayalı görsel renk seçim sistemi sonuçlarda büyük değişimlere sebep olmaktadır. Metamerizm, aydınlatma, aydınlatmanın türü (tungsten veya floresan ampuller), renk illüzyonu ve renk algılama farklılıkları renk seçiminde tutarsızlıklara yol açan etkenlerden bazılarıdır. Geleneksel renk seçimindeki klinik yetersizlikler ortaya çıktığında, daha uygun yöntemlerin geliştirilmesi için çalışmalar başlamıştır. Diş hekimliğinde objektif renk seçimi standartları ihtiyacı doğmuştur, ki böylelikle hata kaynaklı tekrarlanan seanslarda harcanan zaman ve üretim kaybı engellenebilir. Bu amaçla geliştirilmiş spektrofotometre ve kolorimetreler diş hekimliği kullanımına sunulmalarından bu yana bir çok araştırmada kullanılmalarına rağmen klinik kullanımda rutin olarak kullanılmamışlardır. Hassasiyetlerinden dolayı pratik uygulamadaki zorlukları ve yüksek maliyetleri konu hakkında farklı alternatifleri gerektirmektedir.

Dolayısıyla bu çalışmada mevcut elektronik renk belirleme yöntemlerine alternatif bir sistem geliştirilmiştir. Bu sistemde bilgisayara bağlanmış bir kameradan alınan görüntü üzerinde gerçek zamanlı ve durdurulmuş görüntü olmak üzere iki yöntem düşünülmüştür. Temel olarak bu yöntemlerde görüntü üzerinde filtreleme yapılmış, aydınlatma farklarının olumsuz etkisini azaltmak amacıyla HSV renk uzayına dönüştürülmüştür. Elde edilen ton ve doygunluk değerlerinin ayrıştırılıp ortalaması alınmış ve kalibrasyon aşamasında referans alınan değerlerle karşılaştırılmıştır.

SUMMARY

TOOTH COLOUR DETERMINATION BY ELECTRONIC SYSTEMS

Tooth shade determination in aesthetical dentistry, can be done by 2 ways; visual and instrumental. Visual determination depending the dentist's subjective comment, causes significant variations. Metamerism, illumination, type of illumination (tungsten or fluorescent bulbs), color illusion and color sensation are some factors on this incoherency. Researchs for more consistent methods has started, when deficiencies are found out with the traditional shade determination. Objective shade determination standarts are needed to prevent time and production waste caused by repeated clinics due to mistakes. To solve this problem, developments as spectrophotometers and colorimeters have been used in many researchs but not in routine clinic usage. Practical usage difficulties due to their sensitivities and high costs have required alternative researchs.

Therefore, we developed a system as an alternative for existing shade determination systems. In this system, we considered two methods, as real time and still image processing on frames that have been taken from a camera plugged to the computer. Basicly in these methods, frames have been processed by median filtering and converting to HSV color space to minimize suffers from illumination variations. Then, obtained average hue and saturation values have been compared to reference values that have been recorded on the calibration session.

1. GİRİŞ

Estetik diş hekimliğinde kullanılan malzemelerdeki renk uyumu, yapılan operasyonun başarılı olmasında oldukça önemli bir rol oynamaktadır. Renk uyumu, hastanın doğal dişi ile yapılan protezin mümkün olduğu kadar benzer olması ile sağlanmaktadır [1].

Renk seçiminde uygulanan iki yöntem vardır. Görsel olarak çıplak göz ile değerlendirme ve aletsel olarak spektrofotometre, kolorimetre, RGB cihazı gibi cihazların kullanımınıdır. Genelde renk seçimi özel üretilmiş renk skalaları yardımıyla görsel olarak yapılmaktadır. Ancak bu yöntem güvenilmeyen ve yetersiz sonuçlara sebep olabilmektedir. Bu sistem ile renk seçimi oldukça zor ve sübjektiftir. Renk seçiminde kullanılan ışık kaynağının türü, hekimin deneyimi, yaşı ve göz yorgunluğu gibi değişkenler hatalı sonuçlara sebep olabilir. Bu yüzden diş rengi seçiminde kolorimetreler, spektrofotometreler, dijital renk ölçücüler gibi bazı elektronik sistemler geliştirilmiştir [2].

Spektrofotometreler rengin geçirgenliğini, yansımalarını ve emilimini ölçmek için kullanılan bir yöntemdir. Spektrofotometrik renk ölçümleri ölçüm moduna ve kullanılan ışık kaynağına bağlı olarak değişebilir. Bazı spektrofotometreler Specular Component Included (SCI) ve Specular Component Excluded (SCE) olmak üzere iki farklı ölçüm yöntemine göre kullanılabilirler. Standart ışık kaynaklarının bir çok türü de dental malzemelerin renk ölçümlerinde kullanılırlar [3].

Kolorimetreler ise standart bir renk kalibrasyonuna dayanarak rengi tespit edilecek nesnedeki renk verilerini değerlendiren cihazdır. Diş hekimliğinde, renk değerlendirme amacıyla dizayn edilen ilk cihaz 1980'li yılların başlarında Cromascan (Sterngold ,Stamford,Conn) olarak piyasaya sunulmuştur. Fakat sınırlı hassasiyeti ve kullanım zorluğu nedeniyle başarılı olamamıştır. Bu cihazlar üç uyaranlı x,y,z değerlerini veya CIE L*,a*,b* değerlerini verirler. Bu değerlerler matematiksel olarak analiz edilebilir.

Kolorimetrelerle ilgili yapılan çalışmalarda, fotoelektrik üç uyaranlı kolorimetre ile diş hekimliğinde kullanılan porselen örnekler kullanılarak doğru ölçümler elde edilmiştir. Ancak bu cihazlar kusursuz değildir. Diş hekimliğinde kullanılan kolorimetreler ile

diş ve porselen gibi farklı optik özelliklere sahip nesnelerin ölçümü, sonuçları değiştirebilir. Seghi [4] kolorimetre ile topladığı verilerin porselen translüensliği ile değişebileceğini ifade etmiştir. Haywood ve ark. [5] ise kolorimetrelerin diş gibi eğimli ve translusens objelerin renk tespitinden çok, düz yüzeylerde kullanılmaya uygun olarak dizayn edildiği fikrini savunmuşlardır. Bununla beraber bazı firmalar ürettikleri renk tespit cihazları ile beraber firmanın önerdiği porselen sistemlerinin kullanımı ile hatasız sonuçlar elde edilebileceğini iddia etmişlerdir.

Spektrofotometre ve kolorimetreler diş hekimliği kullanımına sunulmalarından bu yana çok sayıda araştırmada kullanılmalarına rağmen klinik kullanımda rutin olarak yaygınlaşmamışlardır. Bu cihazların kullanımı ile yapılan renk seçimi göz ile yapılan renk seçimine göre avantaj sağlamaktadır. Çünkü bu ölçümler objektiftir ve daha hızlı elde edilebilir [2].

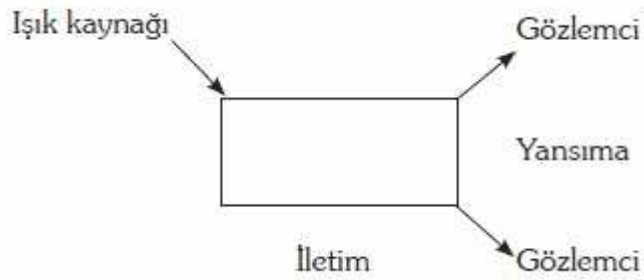
Bu çalışmada, mevcut elektronik renk belirleme yöntemlerine alternatif olarak iki farklı kamera ile gerçek zamanlı ve durdurulmuş görüntüyü işleme yöntemleri kullanılarak renk belirlenmesine yardımcı olmak amaçlanmıştır.

2. GENEL KISIMLAR

2.1. DIŐ HEKİMLİĐİNDE RENK SEĐİMİ

Estetik diő hekimliĐi, esasında diő hekimi ve teknisyenin sanatsal yeteneĐine baĐlı olmasına raĐmen konuyla ilgili bilimsel prensiplerin bilinmesi de son derece önemlidir. Estetik görünümünü etkileyen önemli faktörler renk, şeffaflık, parlaklık ve ışıma (floresanslık) özelliĐidir. Bu faktörlerin her biri aőaĐıdaki deĐiőkenlerden etkilenir [1]:

1. Iőık kaynaĐı
2. Optik parametreler
3. Gözlemci



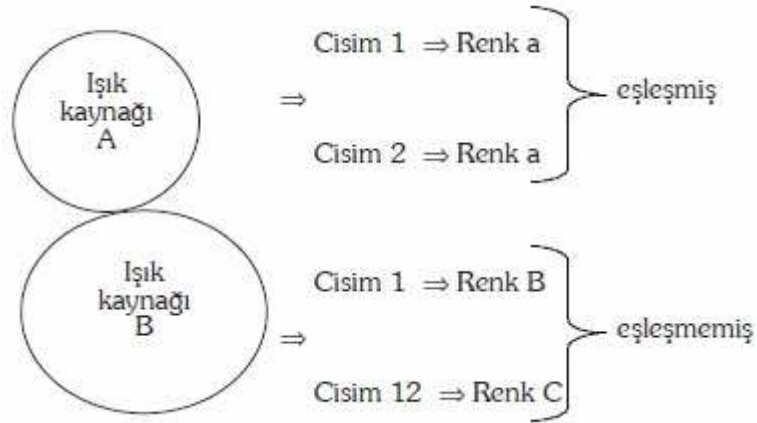
Őekil 2.1. Nesnelerin görülebilmesi için ışık yaymaları veya yansıtmaları [1]

Restoratif malzemelerde bu faktörlerin anlaşılması estetik restorasyonun seĐimi ve yapımında büyük oranda yardımcı olur.

Bir cismin görülebilmesi için ışık yayması veya bir ışık kaynaĐından gelen ışıĐı yansıtması gerekir. Cismin gelen ışıĐa gösterdiĐi tepki belli dalga boylarını emip belli dalga boylarını yansıtmak şeklindedir. Bir cisimden gelen ışık gözün retina tabakası üzerine odaklanıp sinir tepkisi haline dönüşür ve daha sonra beyne aktarır. İnsan gözü 400-700 nm dalga boylarına sahip ışıklara hassastır. Gözün tek bir renk tarafından daimi uyarılması göz yorulmasına ve gözün tepkisinin azalmasına sebep olur. Bununla beraber, renk algılayıcı bölgelerin herhangi bir yerinde olan bozukluk belli renklerin körlüĐüne sebep olur. Bu faktörlerden ötürü her kiőinin renk ayırımı aynı olmayabilir [1].

2.1.1. Metamerizm

Çevresel faktörler ve aydınlatma renk seçiminde önemli rol oynamaktadır. Bir ışık altında aynı renkte gibi görünen cisimler farklı ışık altında farklı görülebilir (Şekil 2.2). Bu olay metamerizm olarak adlandırılır. Bu nedenle renk seçimi biri güneş ışığı altında olacak şekilde, en az üç-dört farklı ışık altında yapılmalıdır. Ayrıca çalışma ortamlarında bir aydınlatma standardizasyonu sağlanmalıdır. Genellikle hastanın ortamına benzer bir ortamda renk seçimi metamerizm etkilerini azaltır. Renk tespiti için en uygun zaman 12.00-15.00 saatleri arasındadır. Gün içindeki zaman, farklı mevsimler ve hava şartları gün ışığının rengini etkiler, bu yüzden standart bir gün ışığı mevcut değildir. Sabah ve akşam gün ışığı, gün içine oranla daha kırmızıdır. Işık kaynağı değiştiğinde, cisimden yansıyan ışık değiştiğinden renk farklı algılanacaktır. Renk ısısı, spektral yansıma eğrisi ve Color Rendering Index (CRI) gibi parametreler standart bir gün ışığı elde etmek amacıyla kullanılmaktadırlar [1].



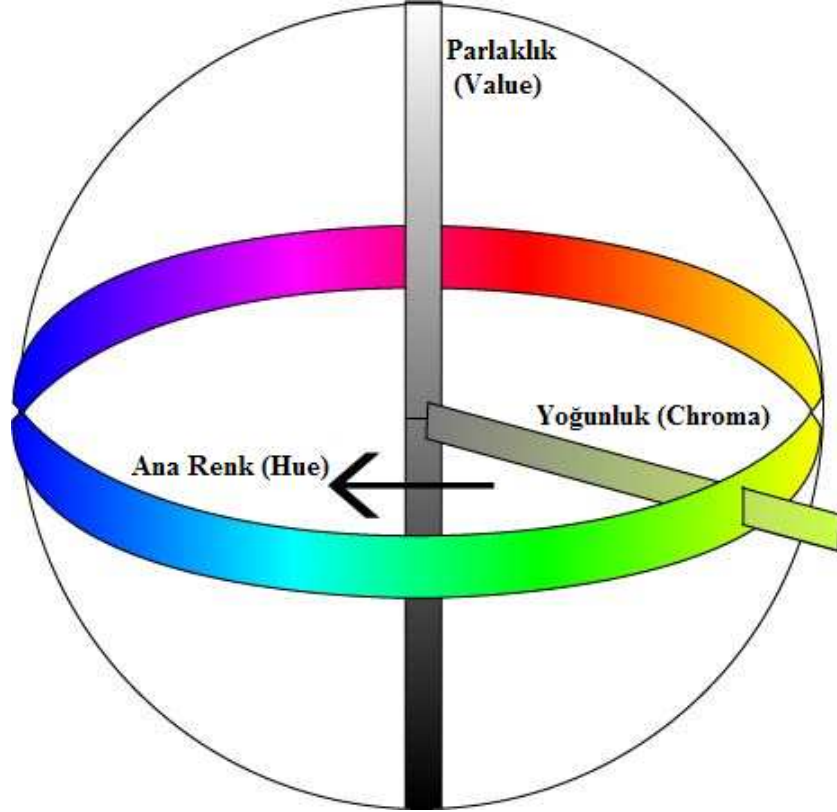
Şekil 2.2. Farklı kaynaklar altında farklı algılanan renkler [1]

2.1.2. Diş Hekimliğinde Kullanılan Renk Sistemleri

Diş hekimliğinde Munsell renk sistemi ve CIE renk sistemi olmak üzere iki renk sistemi kullanılmaktadır [1].

Munsell Renk Sistemi diş hekimliğinde en çok kullanılan renk tanımlama sistemidir. Bu sistemde rengin üç sıfatı; ton veya ana renk (hue), parlaklık (value) ve yoğunluk (chroma) olarak adlandırılır. Value silindirin ortasından geçen dikey eksen üzerinde

gösterilir. Aşağıdan (siyah) başlayarak yukarı (beyaz) kadar gri rengin tonlarını temsil eder. Hue bu silindirin çevresinde düzenlenip, beş ana renge (mor, mavi, yeşil, sarı, kırmızı) ve beş ara renge (mor-mavi, mavi-yeşil, yeşil-sarı, sarı-kırmızı, kırmızı-mor) ayrılmaktadır. Chroma ise yatay yönde ve merkezden dışarıya doğru artmaktadır (Şekil 2.3 ve 2.4).



Şekil 2.3. Munsell renk sistemi

Ana renk (Hue): Bir cismin ana rengi kırmızı, mavi, yeşil gibi herhangi bir renk olabilir.

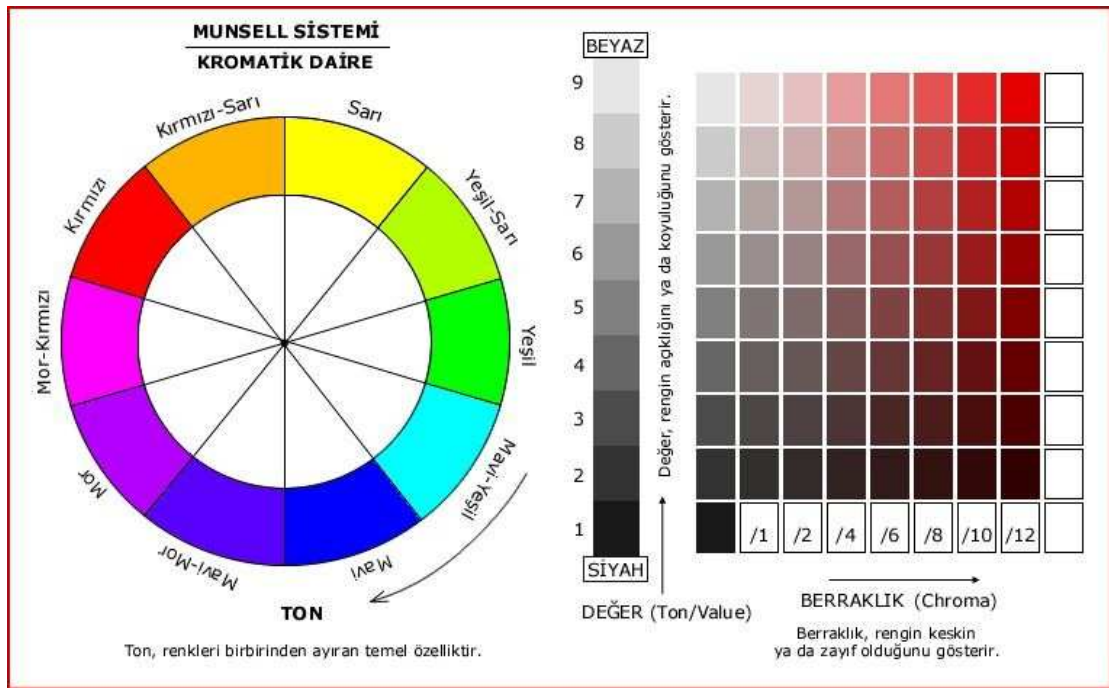
Parlaklık (Value): Parlaklık, bir cisimden yansıyan ışığın miktarıdır. Munsell renk sisteminde parlaklık siyah-beyaz bir skala olarak gösterilmiştir. Parlak cisimler daha az gri iken, düşük parlaklığa sahip cisimler ise fazla miktarda griye sahiptir ve daha koyu görünür.

Yoğunluk (Chroma): Ana rengin doygunluğunu veya yoğunluğunu ifade eder. Yoğunluk ve parlaklık ters orantılı olup, yoğunluk arttığında parlaklık azalır [1].

2.1.3. Dişlerin Translusens ve Floresans Özellikleri

Dişlerde renk bileşenlerinin dışında önemli olan diğer bir özellik de translusensidir. Dişler farklı translusens değerlerine göre karakterize edilir. Translusensi, şeffaf ve opak arasında bir derece olarak ifade edilebilir. Genellikle translusensinin parlaklık ile ters orantılıdır [2].

Floresanslık ise bir malzeme tarafından ışığın emilimi ve sonra daha uzun bir dalga boyunda yayılması olarak tarif edilebilir. Doğal dişler floresans özellik gösterir. Bir protez doğal dişten farklı bir floresansa sahip olduğunda renk uyumsuzluğu oluşur. Floresans özellikli protezler daha parlak ve canlı görüldüğünden, diş hekimliğinde protezin yapısına floresans özelliğe sahip malzemeler eklenmektedir [2].



Şekil 2.4. Munsell renk sisteminin renk tonlarına göre gösterimi

2.1.4. Dişlerin Renk Özellikleri

Doğal bir dişte mine tabakası çok opaktır. Mine daha fazla organik bileşene sahiptir, daha az mineralizedir ve mine kristalleri arasındaki mesafe yüksektir. Bu durum yoğunluğun artmasına sebep olur. Bu yüzden özellikle genç dişlerde mine ışığı

yansıtma eğilimindedir. Dentindeki renk yoğunluğu, minenin baskın etkisi sebebiyle daha az olup, zamanla mine tabakası aşınarak dentini daha belirgin hale getirir. Minenin kalınlığı kole bölgesinde az, kesici kenarda fazladır. Bu yüzden kole bölgesinde yoğunluk en fazlayken, kesici kenara doğru azalmaktadır [2].

Dişlerde en az parlaklık kole bölgesinde ve daha sonra kesici kenardadır. Parlaklığın en fazla olduğu bölge dişin orta üçlüsüdür.

2.1.5. Renk Skalaları

Renk skalaları diş hekimliğinde görsel referans olarak kullanılırlar. Çevre koşulları (aydınlatma türü ve miktarı, göz yorulması, uygulama saati gibi etkenler renk tespitinde rol oynamaktadır. Miller [6] Vita klasik renk skalasının doğal diş örneklerine göre düşük doygunluk ve yüksek parlaklığa sahip olduğunu göstermiştir. Geleneksel renk skala sistemlerinin üretiminde de bazı tutarsızlıklar mevcuttur. Örneğin aynı üreticiden çıkmış birkaç skala üzerindeki renkler birbirinin tamamen aynısı olmayabilir [3].

Vita Classical, Ivoclar Chromascop ve Vitapan 3D Shade Master en yaygın renk skala sistemleridir. Vita Classical renk skala sisteminde, renk tonu harfler ile belirtilmiştir (Şekil 2.6) :

- A - Turuncu
- B - Sarı
- C - Sarı/Gri
- D - Turuncu/Gri (Kahverengi)



Şekil 2.5. Vita Classical renk skalası

Ivoclar Chromascop Sistemi (Şekil 2.7) ana renkleri göstermek için numaralar kullanır:

- 100 - Beyaz
- 200 - Sarı
- 300 - Turuncu
- 400 - Gri
- 500 – Kahverengi



Şekil 2.6. Ivoclar Chromascop renk skalası

3D ShadeMaster ise Vita firmasının yakın geçmişte çıkarmış olduğu, geleneksel skalalardan farklı özel bir sistemdir (Şekil 2.8). Temelde Miller'in [6] yol gösterici çalışması ile dizayn edilmiştir.

- L - sarı ton
- M - orta ton
- R - kırmızı ton

Skala sistemlerinde doygunluk numaralar ile tanımlanmaktadır. Düşük numaralar düşük doygunluğu ifade eder:

- Vita Classical: 1-4
- Chromascop: 10-40
- 3D ShadeMaster: 1-3

Vita Classical ve Ivoclar Chromascop sistemlerinde parlaklık ve doygunluk beraber ifade edilmektedir. Numara arttıkça doygunluk artar fakat parlaklık azalır. 3D ShadeMaster ise parlaklığı ayrı bir numarayla belirtir ve renk skalasında ilk olarak parlaklık değeri gösterilir. 1-5 aralığında numara arttıkça parlaklık azalır. [3]



Şekil 2.7. Vita 3D Master renk skalası

2.1.6. Renk Seçimi

Bir cismin algılanan rengi, cismin fiziksel özelliklerinden, altında bulunduğu ışığın türünden, ortamdaki diğer nesnelere olan ilişkisinden (arka plan) ve kişinin subjektif değerlendirmesinden etkilenir. Bir diş, farklı kişilere ve/veya farklı ışık kaynakları altında farklı renkte görünebilir. Bu yüzden renk seçiminde öncelikle hastanın bulunduğu ortamdaki aydınlatmanın türü önemlidir. Laboratuvar ve muayenehanede aynı ışık kaynakları kullanılmalıdır [2].

Yapılan çalışmalarda 12:00-15:00 saatleri arasının renk seçimi için en iyi zaman olduğu tespit edilmiştir. En iyi renk ısı ve renk kalitesi elde edilmektedir [2].

Günün farklı saatlerinde değiştiği için güneş ışığı aydınlatma ve renk seçimi için her zaman ideal değildir, ayrıca yapay bir aydınlatma gerekli olabilir. En yaygın yapay ışık kaynakları akkor flamanlı (tungsten) ve floresan lambalardır. Akkor ışık kaynaklarında sarı ve kırmızı renkler, floresan ışık kaynaklarında ise mavi renk baskındır.

Beyaz ışığı meydana getirmek için gerekli olan bütün ana renkleri (kırmızı, mavi, yeşil) eşit miktarda içeren ışık 5500 K renk sıcaklığına sahip olup ideal ışık kaynağı olduğu varsayılır. Çalışma alanlarında ışık kaynağı olarak rengi düzeltilmiş floresan lambalar kullanılmalıdır ve renk sıcaklığı 5500 K olmalıdır.

Aydınlatma koşullarının yoğunluğu da etkilidir. Işığın miktarı çok az veya çok yüksek göz rengi zor algılar. Genellikle kliniklerdeki tavan aydınlatmaları yeterince yoğun değildir. Bu yüzden ek aydınlatma olarak ilk önce dental ünite ışıklarına başvurulsa da bu ışıklar renk seçimi için kullanılmamalıdır. Dental ünite ışıklarının çoğu kırmızı-sarı baskın ışık verirler ve gözlerin kamaşmasına sebep olup gözleri yorar. En ideal renk seçimi için 800-2700 lux arasında aydınlatma sağlayan ışık kaynakları kullanılmalıdır.

Doğal diş ve protez yüzeyine gelen ışığın çarptıkları yüzeylerin özelliklerine göre farklı davranışları rengi etkilemektedir. Porselen protezin üzerine gelen ışığın saçılma, kırılma, geçirgenlik, yansıma ve emilim gibi fiziksel özellikleri restorasyonun rengini etkiler. Doğal dişler ve tam seramik protezler düzgün ve yaygın ışık geçirgenliğine izin verirken, metal destekli porselen protezler düzgün ve yaygın yansımaya izin vermektedir. Porselenin camsı yapısı ile yapısındaki kristallerin yoğunluğundan dolayı yansıma ve kırılmaya etki ederek yoğunluğu değiştirmektedir.

Porselen protezlerin klinik uygulamasında uyarlamalar için genellikle yüzeysel boyalar kullanılır. Porselen protezlerde uygulanan dış boyama, doğal diş görünümünü ve diş yüzeyinde yansımalara sebep olur. Ayrıca canlı renkler ile yapılan uygulamalar metamerizme sebep olabilir. Bu yüzden renk seçimi bir kez güneş ışığı altında olmak şartı ile 3-4 değişik ışık kaynağı altında yapılmalıdır.

Porselen protezlerde yüzey yapısı çok önemlidir. Çok düz bir yüzey ışığın düzgün yansımaya neden olur ki, bu da restorasyona camsı bir görüntü verir. Oysa düzensiz bir yüzey ışığın yaygın yansımaya neden olur ve restorasyonlara daha doğal bir görünüm kazandırır [2].



Şekil 2.8. Renk skalası kullanılarak hasta dişine uygun renk seçimi

2.1.7. Renk Seçiminde Genel Prensipler

“Renk seçiminde kullanılan sistem ne olursa olsun, bazı prensiplere uymak gerekir.

1. “Hasta, hekimin göz seviyesinde olmalıdır. Böylelikle retinanın renge en duyarlı kısmı uyarılır.
2. Rengi tespit edilecek olan diş yüzeyi temiz olmalı ve yüzeydeki lekeler temizlenmelidir.
3. Muayenehanenin veya laboratuvarın duvarlarının rengi, rengin algılanmasında önemli faktördür ve algılamayı değiştirebilir. Nötral gri bunun için iyi bir renktir ve gözü dinlendirir.
4. Renk seçimi yapılırken skalalardaki renk örneğinin kesici kenarıyla dişin kesici kenarı baş başa yerleştirilir. Böylece, renk örneklerinin etkili bir şekilde dişlerden izole edilmesi sağlanarak birbirlerinin renklerini yansıtılmaları önlenir.
5. Güneş ışığının değişkenliği sebebiyle pencere güneşlikleri kullanılabilir. Bu durumda aydınlatma 5500 K’lik renk düzenleyici bir ışık kaynağıyla yapılmalıdır. Hastanın elbisesi gri bir önlükle kapatılmalıdır.
6. Parlaklık (value) rengin belirlenmesinde en önemli boyuttur.
7. Renk seçimi asla dental ünit ışığında yapılmamalıdır.
8. Renk seçimi tedavi öncesinde yapılmalıdır. Aksi takdirde tedavi sırasında dehidratasyon meydana gelir ve bu, parlaklığın artmasına sebep olurken yoğunluk ve translusensinin azalmasına sebep olur.

9. Göz yorgunluğu sebebiyle ilk ölçümler daima en iyisidir. Dişlere 5 sn. den fazla bakılması ana renkte sapmalara sebep olur.
10. Vita renk skalasına göre A1 ve B1 renklerinde yoğunluk çok düşüktür ve renk seçimi sırasında bu iki rengi ayırt etmek zordur.
11. Eğer ana rengi belirlemede kuşkuya düşülüyorsa restorasyon A grubu bir renk seçilerek yapılmalıdır. Çünkü dişlerin % 80'i A grubuna aittir.
12. Renk seçimi yapılırken farklı yönlerden bakılarak ışığın farklı açılardan nasıl yansıdığı belirlenmelidir” [2].

Renk şemalarının kullanımı teknisyen ve hekim arasındaki iletişim eksikliğini azaltacaktır.

Dişlerin rengi, dentinin rengi ve onun üzerini kaplayan minenin kalınlığıyla belirlenir. Dentinin rengini belirlemek renk seçiminin ilk ve en önemli bölümüdür. Dentinin parlaklık ve yoğunluk değerleri ise minenin kalınlığına bağlıdır.

Renk seçiminde ve renklendirmenin yapımı sırasında belli kurallara uyularak çalışıldığında hekim ve teknisyen arasında daha iyi iletişim kurulabilir ve bunun sonucu olarak hastayı tatmin edecek daha estetik sonuçlar alınabilir. Doğal dişin formuna ve rengine uygun çalışma yapılması hastayı mutlu edecektir [2].

2.1.8. Görsel Renk Seçiminin Dezavantajları

“Görsel renk analizinin üç dezavantajı vardır:

1. Renk skalalarındaki mevcut renkler yetersizdir.
2. Diş hekimleri arasında ve aynı bireyde günün farklı saatlerinde seçilen renkte tutarsızlıklar olmakta, standardizasyon sağlanamamaktadır.
3. Elde edilen sonuçları CIE renk sisteminde göstermek olanaksızdır” [2].

Görsel renk seçimi subjektif olduğu için her zaman tutarlı sonuçlar almak zordur. Diş hekimleri arasında farklılıklar olduğu gibi, aynı diş hekiminin farklı zamanlarda aynı rengi farklı şekilde algıladığı da bildirilmiştir. Rengin algılanmasına birçok değişken

etki eder. Işık kaynağı, hastanın kıyafeti ve makyajı, renk seçiminin yapılacağı odanın rengi ve dışın yüzey özellikleri rengin algılanmasına etki eden faktörlerdir [2].

Nakagawa, Maruyama ve Shimofusa (1975) tarafından yapılan çalışmada üç diş hekiminden geleneksel renk skalalarını kullanarak renk seçmeleri istenmiştir. Vakaların %86'sında üç diş hekimi aynı renk üzerinde anlaşamamış, yalnızca %14'ünde tüm diş hekimleri aynı fikirde olmuşlardır. Bu sonuçlar görsel renk seçimindeki bireysel farklılaşmayı bariz biçimde ortaya koymaktadır [3].

2.1.9. Diş Rengi Seçiminde Teknolojik Gelişmeler

Geleneksel renk skalaları kullanılarak yapılan görsel renk tespitine alternatif olarak spektrofotometre veya kolorimetre cihazlarıyla renk seçimi yapılabilir [2].

Spektrofotometre, rengin geçirgenliğini ve yansımalarını ölçme prensibiyle geliştirilmiş bir renk tespit cihazıdır. Yapılarında prizma veya dağıtıcı parçalar içeren spektrofotometrelerin renk ölçümleri, ölçüm şekline ve kullanılan ışık kaynağına bağlı olarak değişebilir. Diş hekimliğinde standart ışık kaynağının bir çok türü renk ölçümünde kullanılır. Kolorimetre ise önceden belirlenmiş standart bir renk kalibrasyonuna dayanarak renk verilerini analiz eden bir cihazdır.

Diş hekimliğinde, renk tespit amacıyla geliştirilen ilk cihaz 1980'li yılların başlarında Chromascan (Sterngold, Stamford, ABD) olmuş, fakat hassasiyet problemi ve kullanım zorluğu nedeniyle pek başarılı olamamıştır. Bu cihazlar CIE renk sisteminin bileşenleri olan x, y, z değerlerini veya L*, a*, b* değerlerini verirler. Bu değerler analiz edilebilir ve yorumlanabilir.

Priest ve Lindke [7] renk seçiminde objektif ve doğru değerlendirmeyi sağlamak için optik haritalama sistemini tanımlamışlardır.

Paul ve ark. [8] insan dişlerinin renk analizinde görsel ve spektrofotometrik renk analizlerini karşılaştırmışlar ve spektrofotometrik renk analizinin daha uygun ve verimli bir yöntem olduğu sonucuna varmışlardır. Çalışma dahilinde üç klinisyen 10 hastada bir

protez ihtiyacı için diř rengi seçmişler, daha sonra aynı hastaların aynı dişleri standart bir ışık kaynağı kullanılarak spektrofotometre ile ölçülmüştür. Sonuç olarak 10 hastanın 9 tanesinde spektrofotometrenin tespit ettiği renkler ile hazırlanan protezler tercih edilmiş ve bu sistemin protez uygulamalarında etkili olarak kullanılabileceği ifade edilmiştir.

Sim ve ark. [9] bireyler arasındaki renk algılama farkını incelemek için 10 diş teknisyeni, 15 pratisyen diş hekimi, 15 diş hekimliği fakültesi son sınıf öğrencisi ve 10 protez uzmanının yaptığı renk seçimlerini değerlendirmişlerdir. Sonuç olarak bazı renklerde bireylerin algılamalarının değiştiğini tespit etmişlerdir.

Paravina [10] yaptığı araştırmanın sonucunda testte kullandığı cihazın gün ışığında daha iyi renk seçimi yaptığı sonucuna varmıştır.

Diğer bir arařtırmada kadın ve erkeklerin renk algılamaları arasındaki fark incelenmiş, kullanılan ışık kaynağının erkeklerin renk algılamasında önemli bir farka sebep olmadığı, bununla beraber ışık kaynağının farklı olmasının kadınların renk algılamasında farklılıklara sebep olduğu tespit edilmiştir [2].

Diş hekimliğinde spektrofotometre ve kolorimetreler, görsel renk seçiminin dezavantajlarını elimine edebilmek amacıyla kullanılmaktadırlar. Yapılan çalışmalarda doğru ve tekrarlanabilir ölçümler elde edilmesiyle beraber, bu cihazlar da tamamen kusursuz değillerdir. Seghi [4] kolorimetre ile yaptığı çalışmalarda sonuçların porselen translüensliği ile değişebileceğini ifade etmiştir. Haywood ve ark. [5] ise kolorimetrelerin diş gibi eğimli ve translüens objelerden ziyade, düz yüzeylerde kullanılmaya daha uygun olduğunu belirtmişlerdir.

Bazı firmalar renk tespit cihazları ile beraber kendi önerdikleri porselen sistemlerinin kullanımını ile hatasız sonuçlar elde edilebileceğini iddia etseler de, bu yönde yapılan çalışmalarda ise sistemler arasında istatistiksel olarak fark olmadığı sonucuna varılmıştır [2].

2.1.10. Piyasadaki Popüler Renk Seçim Cihazları

İlk olarak 1998 yılında tek nokta kaynaklı bir spektrofotometre kullanan ShadeEye-EX Chroma Meter (Shofu, Menlo Park, ABD)'in geliştirilmesiyle renk tespitinde elektronik sistemler ortaya çıkmıştır. Shofu ShadeEye Chroma Meter (Şekil 2.11) sisteminde gözlenen rengi tespit etmek için nokta referans sistemi kullanılmaktadır [3].



Şekil 2.9. Shofu ShadeEye-EX cihazı

Dişler çok renkli yapıda olduklarından ve dolayısıyla daha doğru bir analiz elde edebilmek için bu sistemi kullanırken çok sayıda referans noktasından ölçüm alınması gerekir. Shofu Chroma Meter'in diğer bir dezavantajı da Shofu porselen sistemi ile sınırlı uyumda olmasıdır.

Robert [11] dişleri görmek ve renklerini analiz etmek için bilgisayar destekli sayısal video kamera görüntüsü teknolojisi kullanan ShadeScan (Cynovad, Montreal, Kanada)'ın kullanımından bahsetmiştir (Şekil 2.12). Dijital görüntü daha sonra geleneksel renk skalalarının oluşturduğu veritabanı ile karşılaştırılır. Sonuç raporu, translüsenesi ve yoğunluk kontrast alanlarını gösteren bir renk dağılım haritası da içerir. Doğal diş ile renk haritası arasındaki değer farklılıkları ile ilgili bilgiler de detaylı olarak verilir.



Şekil 2.10. Cynovad Shadescan cihazı

Cihazın kullanımını kolay olup, görüntü elde etmek için aygıt dış yüzeyine dik veya en fazla 20 derecelik açıyla tutulur. Yüzey yansımalarını en aza indirmek için dış yüzeyini 45 derece açı ile aydınlatan halojen ışık kaynağı mevcuttur. Sistemin kritik konusu, görüntü alımı ve analizini standardize eden devamlı ve sabit bir aydınlatma düzeyidir. Sabit aydınlatmayı garantilemek için ise çıkış gücü devamlı kalibre edilmektedir.

Diğer bir renk tespit sistemi olan SpectroShade System (MHT International, Newton, ABD) ilk olarak 2001 yılında Cherkas [12] tarafından tarif edilmiştir (Şekil 2.13). Renk analizi için spektrofotometrik yaklaşımı kullanan sistem, bilgisayar desteğiyle yaklaşık üç yüz bin referans noktasından faydalanır. Kalibrasyonu hassastır ve etkili bir sonuç için iki görüntü gerektirmektedir..



Şekil 2.11. MHT SpectroShade cihazı

Günümüzdeki en popüler renk seçim cihazlarından biri de Vita firmasının üretmiş olduğu EasyShade'dir (Şekil 2.15). Bu cihaz da spektrofotometre mantığıyla çalışır, ölçümleri CIE Lab renk formatında alıp Vita Klasik renk skalasına göre sonuç verir.



Şekil 2.12. Vita EasyShade cihazı

Çalışmasında spektrofotometre kullanan Miller [6], spektrofotometrenin performansının %1'e kadar doğru olduğunu; spektrofotometrenin doğal bir diş üzerindeki ölçüm sırasında küçük kaymalara karşı hassas olduğu sonucunu elde etmiştir. Ayrıca doğal dişler ve porselen protezlerin yüksek yansıtma ve translüensi özelliklerinden dolayı protezin merkez bölgelerinde ölçümler yapılması gerekmektedir. Bir başka renk analiz sistemi olan Shade Vision System (X-Rite Co., Grandville, ABD) renk ölçümünde

kolorimetrik verilerden yararlanmaktadır. Kablosuz el aletiyle, alınan görüntüler, (Şekil 2.14) daha sonra analiz ve değerlendirme için bilgisayara yüklenir [3]



Şekil 2.13. X-Rite ShadeVision cihazı

2.2. SAYISAL GÖRÜNTÜ İŞLEME

Bilgisayar teknolojilerinin ilerlemesiyle günümüzde görüntü işleme uygulamalarında da başarılı çalışmalar yapılmaya başlanmıştır. Bilim ve teknolojinin yapay zekâya sahip güvenilir teknolojiler geliştirebilmesi ses ve görüntünün verimli biçimde işlenebilmesini gerektirmektedir. Gelecek teknolojiler insanın duyu organlarını taklit edebilen ve algıladığı verileri değerlendirip kararlar alabilen teknolojiler olmalıdır [13].

2.2.1. Sayısal Görüntü İşleme Sisteminin Temel Öğeleri

1980'li yıllarda piyasadaki görüntü işleme sistemleri genellikle ana bilgisayarlara bağlı olan çevre birimlerinden oluşmaktaydı. 1990'lı yılların başlarından itibaren ise görüntü işleme sistemleri, kişisel bilgisayarlara takılabilen kartlar olarak geliştirilmiştir [13].

Büyük görüntülerin alındığı ağır görüntü işleme uygulamaları için büyük çaptaki görüntü işleme sistemleri halen yoğunlukla kullanılıyor olsa da günümüzde görüntü işleme sistemlerinin küçülmesi ve aynı zamanda kişisel bilgisayarların bu uygulamalara uyumlu hale getirilmesi amaçlanmıştır. Sayısal görüntü işleme için kullanılan bir sistemin temel bileşenleri Şekil 2.16'da gösterilmektedir.

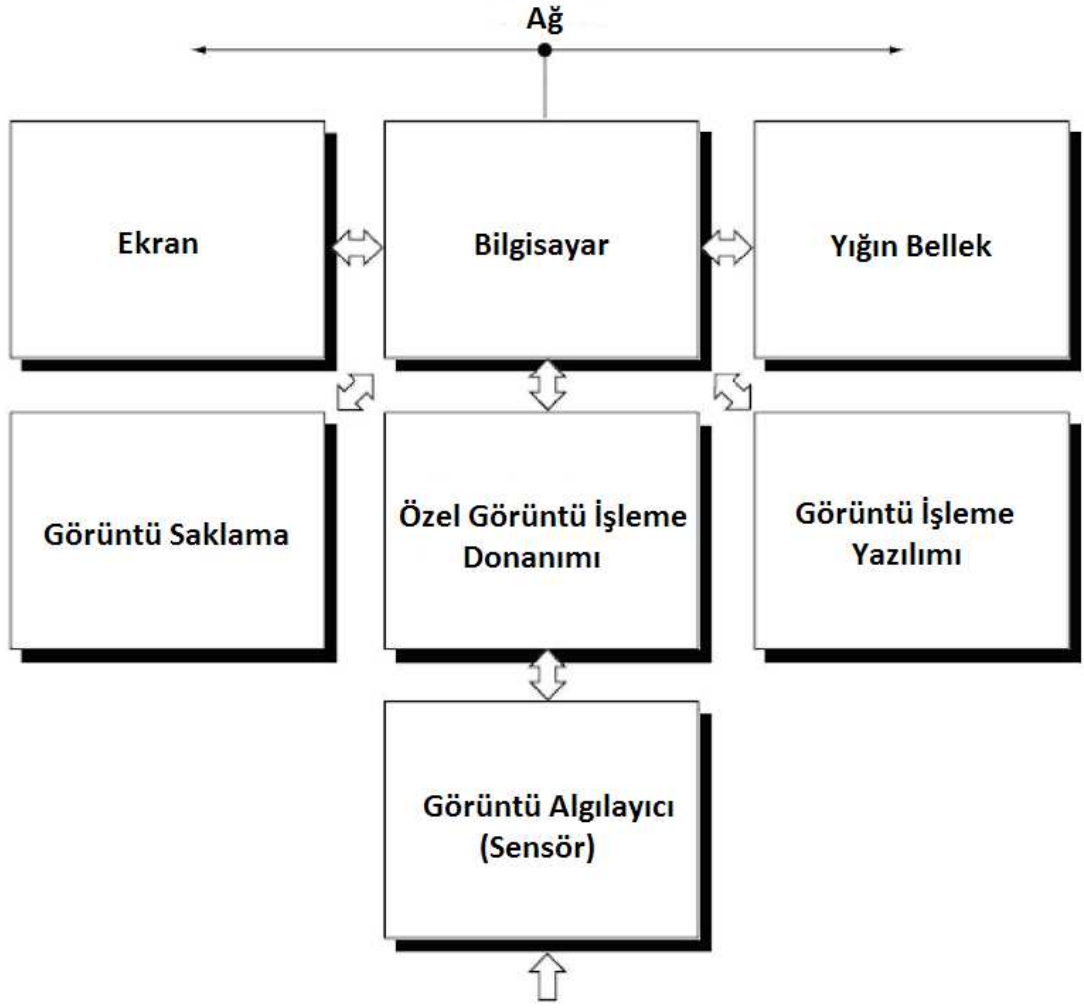
Sayısal resimler elde etmek için görüntülemek istenilen nesneye duyarlı fiziksel bir aygıt (görüntü algılayıcı) ve bu aygıtın çıktısını sayısal forma dönüştüren analog-sayısal çevirici elemanlara (özel görüntü işleme donanımı) ihtiyaç vardır.

Görüntü işleme için geliştirilen yazılımlar belirli görevler yapan özel modüllerden oluşur. İyi tasarlanmış bir yazılım kullanıcının özel modülleri kullandıran kodları yazmasına olanak vermelidir.

Her bir pikselin 8 bit uzunluğunda olduğu 512x512 piksel büyüklüğünde sıkıştırılmamış bir görüntü dosyası 512 kilobyte bellek gerektirir. Çok sayıda resim söz konusu olduğunda bir görüntü işleme sistemi için yeterli bellek sağlamak çok önemlidir. Görüntü işleme uygulamalarında kullanılan sayısal bellek temelde üçe ayrılır.

1. İşlem sırasında ihtiyaç duyulan kısa süreli bellek.
2. Hızlı erişim için çevrimiçi bellek.
3. Arşivsel bellek.

Bellek byte (8 bit) ölçü birimi ile ölçülmektedir. Kısa süreli bellek oluşturmanın bir yöntemi bilgisayarın rastgele erişimli belleğini (RAM) kullanmaktır. Diğer bir yöntem ise frame buffer olarak bilinen özel devreleri kullanmaktır. Bu hafıza modülleri bir ya da daha fazla görüntüyü saklayabilir ve video mantığındaki gibi saniyede 30 kareye ulaşılabilir. Frame buffer metodu görüntü üzerinde ani yakınlaştırma ve uzaklaştırma işlemlerine olanak sağlar. Frame bufferlar genellikle Şekil 2.16'da gösterilen özel görüntü işleme donanımlarının içinde olacak şekilde dizayn edilir. Çevrimiçi bellek genelde manyetik diskler ya da optik medya (CD, DVD) bellekleri şeklindedir. Arşivsel bellek seyrek erişim için, çevrimiçi bellek ise sık erişim için kullanılır [13].



Şekil 2.14. Sayısal görüntü işleme sisteminin temel öğeleri

Ekran olarak günümüzde çoğunlukla renkli CRT veya LCD monitörler kullanılmaktadır. Monitörler bilgisayarlara haricen takılabilen veya anakart üzerine entegre edilmiş grafik işlemcilerin çıkışına bağlanmaktadır.

Görüntü saklamak için yazıcılar, film kameraları, optik medya ortamları gibi sayısal ürünler kullanılır.

Ağ ise artık günümüzde hemen her bilgisayar sisteminde kullanılmakta olup görüntü işleme uygulamalarında büyük miktardaki bilgilerin iletimi için kullanılmaktadır [13].

2.2.2. Sayısal Görüntünün Bazı Temel Kavramları

2.2.2.1. Siyah - Beyaz Görüntü

Siyah beyaz bir görüntü, parlaklığı gösteren iki boyutlu bir fonksiyon olarak düşünülebilir. Bu $f(x, y)$ fonksiyonunun (x, y) koordinatlarındaki değeri resmin o noktadaki parlaklığını gösterir. Siyah beyaz bir resmin (x, y) koordinatlarındaki parlaklığına resmin bu noktadaki gri ton değeri adı verilir [13].

“ $f(x, y)$ 'nin iki bileşeni vardır:

1. Ortamdaki ışığın miktarı (aydınlatma),
2. Nesnenin ışığı yansıtma oranı (yansıtma).

Aydınlanmayı $i(x, y)$, yansıtmayı $r(x, y)$ ile gösterirsek $f(x, y)$ bunların çarpımı şeklinde yazılabilir:

$$f(x, y) = i(x, y) \cdot r(x, y) \quad (2.1)$$

Burada teorik olarak

$$0 < i(x, y) < \infty \quad (2.2)$$

ve

$$0 < r(x, y) < \infty \quad (2.3)$$

kabul edilir. $i(x, y)$ 'yi ışık kaynağı, $r(x, y)$ 'yi nesnelerin yansıtma özellikleri belirler. $r(x, y)$ için 0 limit değeri ışığın tamamen absorbe edilmesi, 1 limit değeri ışığın tamamen yansıtması anlamına gelir. Aydınlanma ortalama olarak açık bir günde 9000, bulutlu günde 1000, tipik bir ofiste 100 foot - kandeladır. Yansıtma için ortalama olarak bulunan bazı değerler en koyu cisim için 0.01, paslanmaz çelik için 0.65, kar için 0.93'tür.

$f(x,y)$ fonksiyonun (x,y) noktasındaki değerine λ dersek (2.1), (2.2) ve (2.3) denklemlerinden λ 'nın

$$L_{\min} < \lambda < L_{\max} \quad (2.4)$$

aralığında değiştiğini görürüz. Teoride tek gereklilik L_{\min} 'in pozitif olması ve L_{\max} 'ın sonlu olmasıdır. Uygulamalarda $L_{\min} = i_{\min} \cdot \gamma_{\min}$ ve $L_{\max} = i_{\max} \cdot \gamma_{\max}$ yazarsak ve yukarıdaki değerleri göz önünde alırsak, kapalı mekanlar için $L_{\min} \cong 0.005$ ve $L_{\max} \cong 100$ bulunur. $[L_{\min} \cdot L_{\max}]$ aralığına gri ton ölçeği (gray scale) adı verilir. Pratikte bu aralık $[0, L]$ aralığına kaydırılır. Burada $\lambda = 0$ siyah ve $\lambda = L$ beyaz olarak kabul edilir. Aradaki değerler grinin tonlarına karşılık gelir” [13].

2.2.2.2. Sayısal Görüntü

Görüntü üzerinde işlem yapabilmek için $f(x, y)$ görüntü fonksiyonun koordinat sisteminin sayısallaştırılması (görüntü örnekleme) ve genliğinin sayısallaştırılması (gri ton nicelenmesi) gerekir. Yani sayısal resim hem düzlemsel koordinatların, hem de parlaklığın ayrı olarak değerlendirildiği bir görüntü fonksiyonudur [13].

$$f(x, y) \cong \begin{pmatrix} f(0,0) & f(0,1) & \cdot & \cdot & f(0, M-1) \\ f(0,0) & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ f(N-1,0) & \cdot & \cdot & \cdot & f(N-1, M-1) \end{pmatrix} \quad (2.5)$$

(2.5) denkleminde $f(x,y)$ fonksiyonu, sürekli bir resmi göstermekte olup denklemin sağ tarafı resmin sayısal olarak ifade edildiği bir matris olarak düşünülebilir. Matriste satır ve sütunlar görüntüdeki bir noktanın koordinatları, bu noktadaki değer de gri ton değerini göstermektedir. Ayrıca matristeki her bir elemana piksel adı verilir. 8 bit uzunluğundaki siyah beyaz görüntü için her bir pikselin değeri 0 ile 255 arasında değişebilir [13].

2.2.2.3. Renkli Görüntü

Sayısal sistemlerde renkli görüntüleri ifade etmek için çeşitli renk uzayları kullanılır. Örneğin en yaygın olarak kullanılan RGB (kırmızı-yeşil-mavi) renk uzayı, görüntü üzerindeki her pikselin bu üç ana rengi ne kadar barındırdığını gösteren bir sistemdir. Bu durumda RGB uzayındaki renkli bir görüntü 3 boyutlu matris şeklinde ifade edilir. Her bir ana rengin 8 bit uzunluğunda olduğu bir RGB görüntü toplam 24 bit uzunluktadır [13].

2.2.2.4. Görüntü Dosyaları

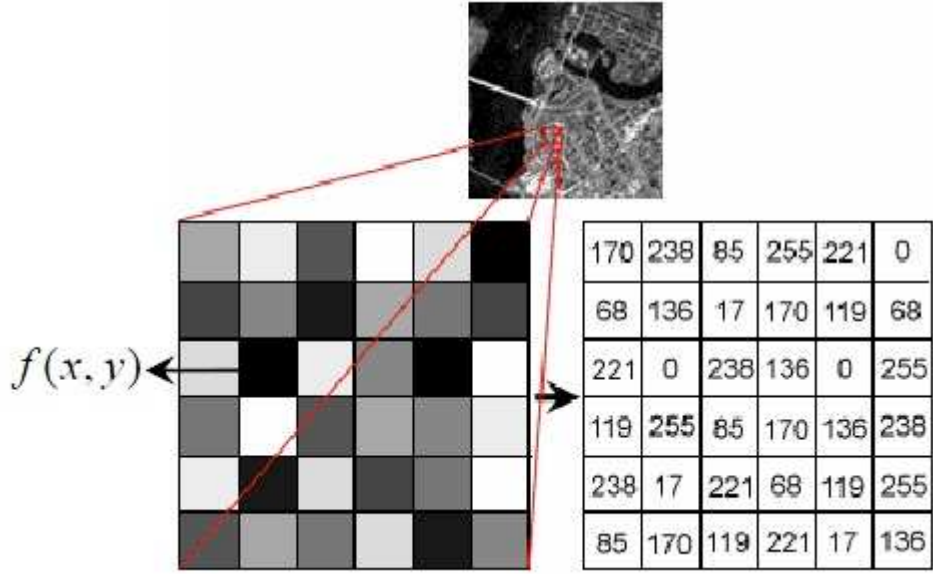
Görüntü işlemede ilk olarak görüntüler alınıp sayısallaştırılır. Sayısallaştırılmış görüntüler ise bilgisayar sistemlerinde video ya da resim dosyası şeklinde saklanır [13].

2.2.2.5. Piksel

Piksel bir görüntüdeki en küçük birim olup, sayısal görüntüler bir araya gelen pikseller topluluğundan oluşmaktadır (Şekil 2.17). Sayısal görüntüler bir matris şeklinde tanımlanır. Matrisin her bir elemanı birer pikseldir. Aksi belirtilmedikçe piksellerin en ve boy oranları eşittir [13].

2.2.2.6. Çözünürlük

Çözünürlük bir görüntü üzerindeki nokta sayısıdır. Örneğin görüntünün sayısal formu olan matrisin satır ve sütun sayısı 512 ise o görüntünün çözünürlüğü “512x512” olarak ifade edilir. Başka bir örnek vermek gerekirse; günümüzde yaygın kullanılan dijital fotoğraf makinelerinde ana unsur olarak göze çarpan “megapiksel (milyon piksel)” kavramı çözünürlüğü ifade eder. Örneğin üzerinde “10 megapiksel” yazan bir dijital fotoğraf makinesinin çektiği bir fotoğrafın üzerinde yaklaşık 10 milyon adet nokta bulunur.



Şekil 2.15. Sayısal görüntü ve piksel [13]

2.2.2.7. Sıkıştırma

Özellikle yüksek çözünürlükteki renkli görüntü dosyaları sıkıştırmasız olarak bellekte büyük miktarda yer kaplarlar. Bu probleme karşı çeşitli sıkıştırma yöntemleri geliştirilmiştir. Bunlardan en çok bilineni JPEG sıkıştırmasıdır.

2.2.3. Görüntü Pekiştirme

Bir görüntünün özel bir uygulama alanı için orijinalinden daha uygun bir hale getirilmesi görüntü pekiştirme (image enhancement) olarak adlandırılır. Görüntü pekiştirme, insana yönelik veya bilgisayara yönelik olmak üzere iki şekilde uygulanabilir. Örneğin görüntü üzerindeki bir nesnenin detaylarının ortaya çıkarılması için uygulanan kontrast düzenlemesi insana yönelik uygulamalara örnek gösterilebilir. Bilgisayara yönelik uygulamalar ise ileri seviye görüntü işleme uygulamalarından önce görüntünün hazır hale gelmesini sağlayan ara basamaklardır [13].

Görüntü düzleminde uygulanan pekiştirme işlemleri, nokta işleme ve maske işleme teknikleri olarak iki kategoriye ayrılır. Bir görüntüyü oluşturulan pikseller kümesine

görüntü düzlemi adı verilir ve görüntü düzlemi yöntemleri bu pikseller üzerinde işlem yapan yöntemlerdir.

Görüntü düzleminde uygulanan görüntü işleme fonksiyonları temel olarak aşağıdaki şekilde gösterilebilir:

$$g(x, y) = T [f(x, y)] \quad (2.6)$$

Bu denklemde $f(x, y)$ giriş görüntüsü, $g(x, y)$ işlenmiş görüntü, T ise f üzerinde (x, y) koordinatlarının komşuluğunda tanımlı bir operatördür.

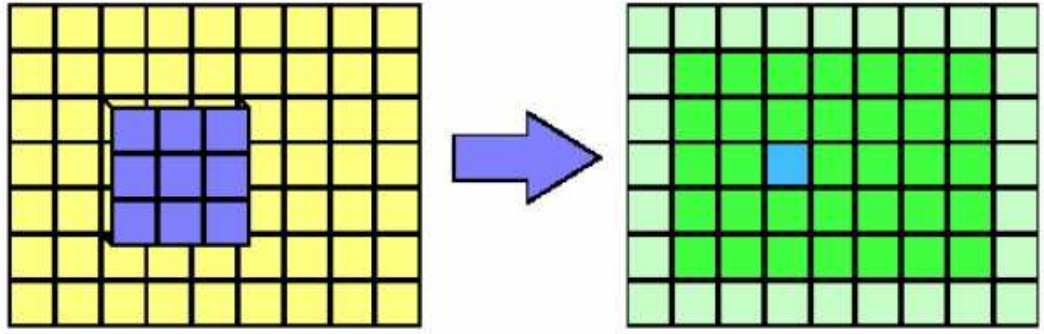
(x, y) 'nin bir komşuluğunu belirlemek için genelde merkezi (x, y) olan dörtgen şeklinde bir bölge tanımlanır. Şayet bu bölgenin boyutları 1×1 olarak belirlenmişse, g 'nin alacağı değer f nin (x, y) noktasındaki değerine bağlıdır. Bu yüzden bu kategorideki teknikler nokta işleme teknikleridir. (x, y) 'nin bir komşuluğunu belirlemek için 3×3 , 5×5 gibi büyük bölgeler de tanımlanabilir. Bu bölgeler maske veya filtre olarak adlandırılır. Uygulama sonucunda g 'nin (x, y) noktasındaki değeri, f nin merkezi (x, y) olan bölgedeki pikseller ve bu bölge üzerine uygulanan filtre matrisinin katsayılarına bağlıdır. Bu uygulamaya ise maske işleme teknikleri denir [13].

2.2.4. Filtreleme

Filtreleme belirli filtre matrislerinin görüntü üzerinde gezdirilerek orijinal görüntünün her piksel değerinin yeniden hesaplanmasıdır (Şekil 2.19). Filtreler orijinal görüntü üzerinde yumuşatma, keskinleştirme, kenar saptama gibi değişik efektler uygulanmasını sağlar.

$h(i, j)$ filtre fonksiyonu olmak üzere filtreleme işlemi şu şekilde ifade edilebilir:

$$f(x, y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i, j) \times f(x-i, y-j) \quad (2.7)$$



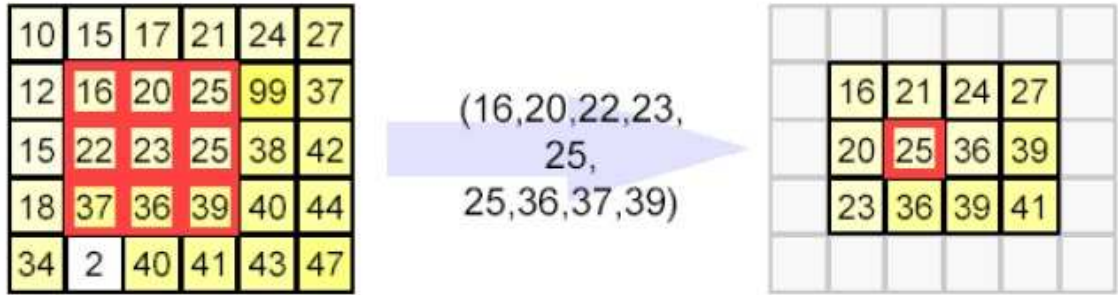
Şekil 2.16. Filtreleme işleminin kabaca gösterimi

2.2.5. Medyan Filtresi

Bir dizideki herhangi bir değerin o dizideki ortanca değer ile değiştirilmesini sağlayan filtredir. İki boyutlu olduğu zaman görüntü için medyan filtresi geliştirilmiş olur [14]. Şekil 2.20'de 3x3 boyutundaki bir örnek medyan filtresi görülmektedir.

$$m(k) = \text{MED } w(k)$$

$$= \text{MED } \{x_{-n}(k), \dots, x_{-1}(k), x_0(k), x_1(k), \dots, x_n(k)\}$$



Şekil 2.17. 3x3'lük medyan filtresi

2.3. RENK UZAYLARI

Renk uzayları, renkleri tanımlamak için kullanılan sayısal sistemlerdir. Renk biliminin temeline göre bir rengi belirlemek için bağımsız 3 değişkene ihtiyaç vardır. Bu yüzden renk uzayları 3 boyutlu olarak tasarlanmaktadır. En popüler renk uzayları bilgisayar grafiklerinde kullanılan RGB, video sistemlerinde kullanılan YCbCr ve genelde

matbaacılıkta kullanılan CMYK'dır. Tüm renk uzayları, kamera, tarayıcı gibi aygıtlardan alınan RGB bilgisinin ilgili formüllerle dönüştürülmesiyle [15].

2.3.1. RGB renk uzayı

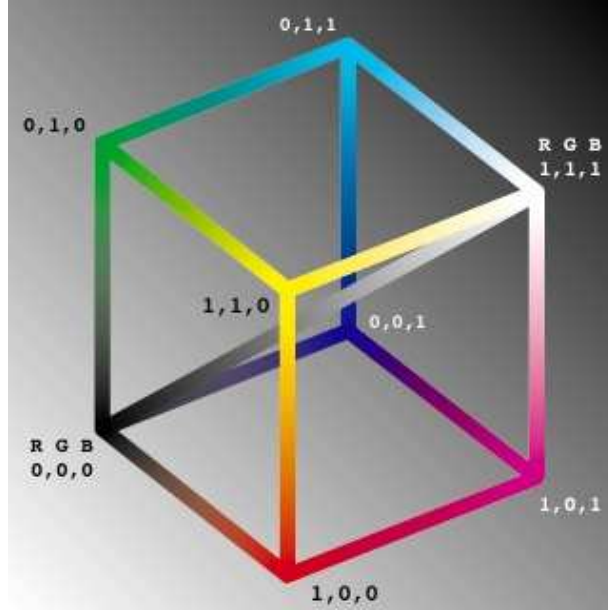
RGB renk uzayı bilgisayar grafikleri üzerinde yoğun bir kullanım alanına sahiptir. Kırmızı (**R**ed) , yeşil (**G**reen) ve mavi (**B**lue) birbirine karıştırılabilen üç temel renk olup birbirlerine karıştırılarak diğer renklerin elde edilmesine olanak sağlar. RGB renk uzayı 3 boyutlu olarak bir küp şeklinde gösterilebilir (Şekil 2.21). Tablo 2.1.'de küpün köşelerindeki seviyelere karşılık ortaya çıkan renkler gösterilmektedir [15].

Tablo 2.1. Küp köşelerinde elde edilen renk tonları

	Aralık	Beyaz	Sarı	Camgöbeği	Yeşil	Eflatun	Kırmızı	Mavi	Siyah
R	0-255	255	255	0	0	255	255	0	0
G	0-255	255	255	255	255	0	0	0	0
B	0-255	255	0	255	0	255	0	255	0

RGB renk uzayı, katot tüplü renkli monitörlerin de renk oluşturmak için kırmızı, yeşil ve mavi rengi kullanmalarından dolayı bilgisayar grafikleri için en uygun renk uzayıdır. Uzun zamandır kullanıldığından çok sayıda yazılım örneği mevcuttur. Bu yüzden tasarım kolaylığı açısından yeni geliştirilen sistemlerde de RGB renk uzayı sıkça tercih edilmektedir [15].

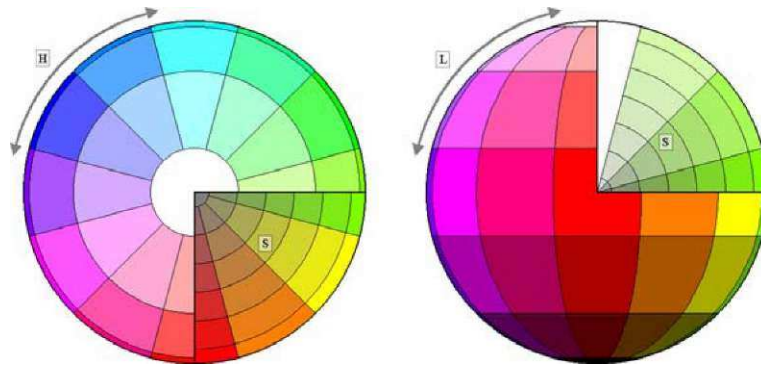
Görüntü oluşturmada yıllardır başarıyla kullanılan RGB renk uzayının görüntü işleme uygulamalarında kullanılması ise etkili bir yol değildir. Örneğin verilen bir pikselin renk yoğunluğu değiştirilmek istendiğinde, bellekten üç renk bileşeni de okunup, bunlar üzerinde işlem yapıp, sonucun tekrar yazılması gerekmektedir. Ayrıca renk ve nesne tespit uygulamalarında RGB değerleri parlaklığın değişiminden belirgin şekilde etkilenmektedir. Görüntüdeki renk ve parlaklık değerlerine doğrudan erişim hakkı sağlayan farklı bir renk uzayı kullanılarak bazı işlem basamakları daha kolay ve hızlı yapılabilir [15].



Şekil 2.18. RGB renk uzayı

2.3.2. HSI, HLS ve HSV renk uzayları

HSI ve HSV renk uzayları daha kolay renk seçimi yapabilmesi amacıyla geliştirilmişlerdir. Renkleri görerek seçilmesi gerektiği durumlarda idealdirler. HLS, HSI ve HSV renk uzaylarında ortak harfler olan H (Hue) renk tonunu, S (Saturation) renk doyumunu ifade eder. Parlaklık ise HLS uzayında L (Luminance), HSI uzayında I (Intensity) ve HSV uzayında V (Value) ile gösterilir. Şekil 2.22 HLS renk uzayını göstermektedir [15].



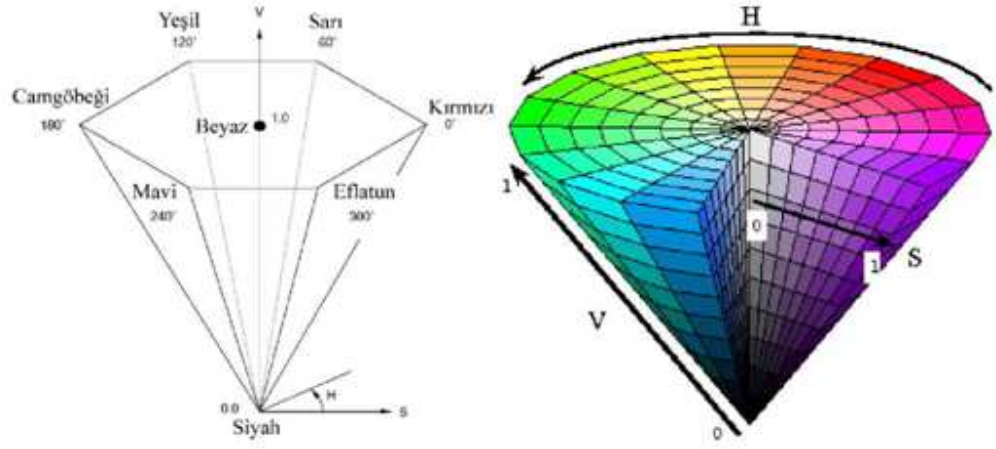
Şekil 2.19. HLS renk uzayı

HSI renk uzayı katsayı eşitleme, histogram gibi geleneksel görüntü işleme uygulamaları için en ideal renk uzayıdır. HSV renk uzayı ise renk doyumu açısından büyük bir aralığa

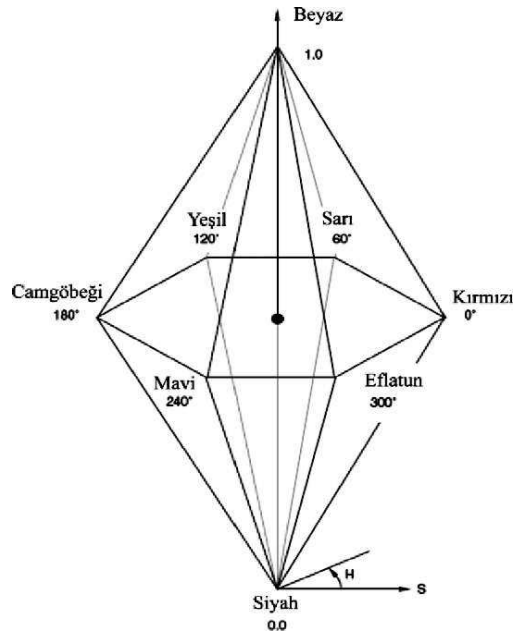
sahip olduğu için, renklerle doğrudan alakalı görüntü işleme uygulamalarında kullanılmaktadır.

HSV renk uzayı altıgen piramit şeklinde gösterilebilir (Şekil 2.23). Piramidin tepe noktası siyah, tabanı beyaz (V) ifade eder. Merkezden dışarı doğru gidildikçe doygunluk (S) artar. Taban boyunca da renk tonu (H) değişir [15].

HSI renk uzayı ise birbirine yapışık çift altıgen piramit şeklinde gösterilir (Şekil 2.24) [15].



Şekil 2.20. Altıgen piramit şeklindeki HSV renk uzayı



Şekil 2.21. Çift altıgen piramit şeklinde HSI renk modeli

2.3.3 RGB Renk Uzayından HSV Renk Uzayına Dönüşüm

$r, g, b \in [0,1]$, $\max = \{ r, g, b \}$ değerlerinin maksimumu, $\min = \{ r, g, b \}$ değerlerinin minimumu ve $h \in [0, 360]$ olmak üzere, RGB uzayından HSV uzayına dönüşüm yapmak için gereken formüller Şekil 2.25'te gösterilmektedir.

$$\begin{aligned}
 h &= \begin{cases} 0 & \text{eğer } \max = \min \\ (60^\circ \times \frac{g-b}{\max - \min} + 360^\circ) \bmod 360^\circ, & \text{eğer } \max = r \\ 60^\circ \times \frac{b-r}{\max - \min} + 120^\circ, & \text{eğer } \max = g \\ 60^\circ \times \frac{r-g}{\max - \min} + 240^\circ, & \text{eğer } \max = b \end{cases} \\
 s &= \begin{cases} 0, & \text{eğer } \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & \end{cases} \\
 v &= \max
 \end{aligned}$$

Şekil 2.22. RGB-HSV dönüşüm formülleri

3. MALZEME VE YÖNTEM

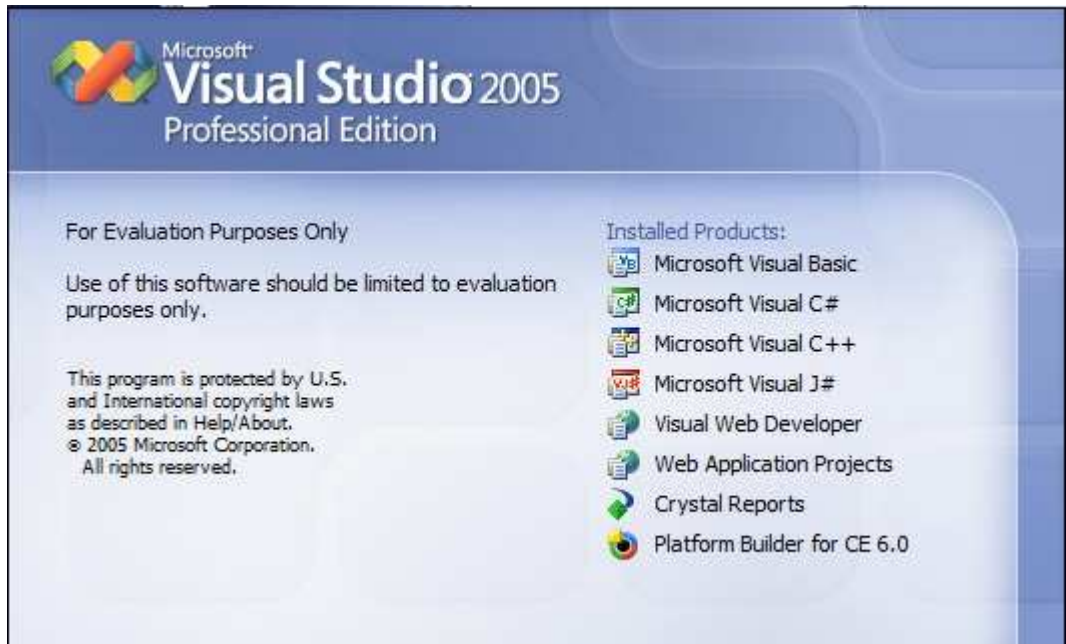
Bu çalışmada aşağıdaki materyaller kullanılmıştır:

- Geliştirme ortamı:
 - PC: AMD PHENOM II X2 550 3.1 Ghz İşlemci – 4 GB DDR3 1600 Mhz RAM
 - Microsoft Visual Studio 2005
 - Intel OpenCV 2.1
- Kameralar:
 - Ağız İçi Kamera
 - Logitech Webcam Pro 9000
- Vita Classic Renk Skalası

3.1. YAZILIM GELİŞTİRME ORTAMININ KURULUMU

3.1.1. Microsoft Visual Studio 2005

Uygulama geliştirme ortamı olarak Microsoft firmasının ürettiği Microsoft Visual Studio 2005 Professional Edition ve C++ programlama dili kullanılmıştır. (Şekil 3.3)



Şekil 3.1. Microsoft Visual Studio 2005

“http://download.microsoft.com/download/e/1/4/e1405d9e-47e3-404c-8b09-489437b27fb0/En_vs_2005_Pro_90_Trial.img” adresinden Visual Studio 2005 90 günlük deneme sürümü indirilip gerekli güncellemeleri yapılmıştır.

3.1.2. OpenCV

OpenCV (Open Computer Vision) ; Windows, Linux, Mac OS X, PSP (PlayStation Portable) platformları üzerinde çalışabilen, C diliyle yazılmış açık kaynak kodlu bir görüntü işleme kütüphanesidir. Intel tarafından ilk sürümü 1999’da çıkarılmış olup, şu anda [SourceForge](http://sourceforge.net) tarafından geliştirilmektedir. İçerdiği fonksiyonların büyük kısmı platformdan bağımsız olarak çalışır [16].

“OpenCV içerisindeki bilgisayarla görüş (computer vision) ve görüntü işleme (image processing) algoritmalarını kullanılarak hareket tespiti, yüz algılama ve tanıma, kamera kalibrasyonu gibi pek çok uygulama yapılabilir. Bir resim veya video doyası kolaylıkla okunabilir, üzerinde işlem yapıp yenileri yaratılabilir. Hatta farklı kameralardan alınmış görüntüleri yorumlayıp, herhangi bir nesnenin konumu belirlenebilir” [16].

OpenCV’nin “Machine Learning” adlı bölümünde yapay zeka ve kendiliğinden öğrenmeyle ilgili fonksiyonlar vardır [16].



Şekil 3.2. OpenCV

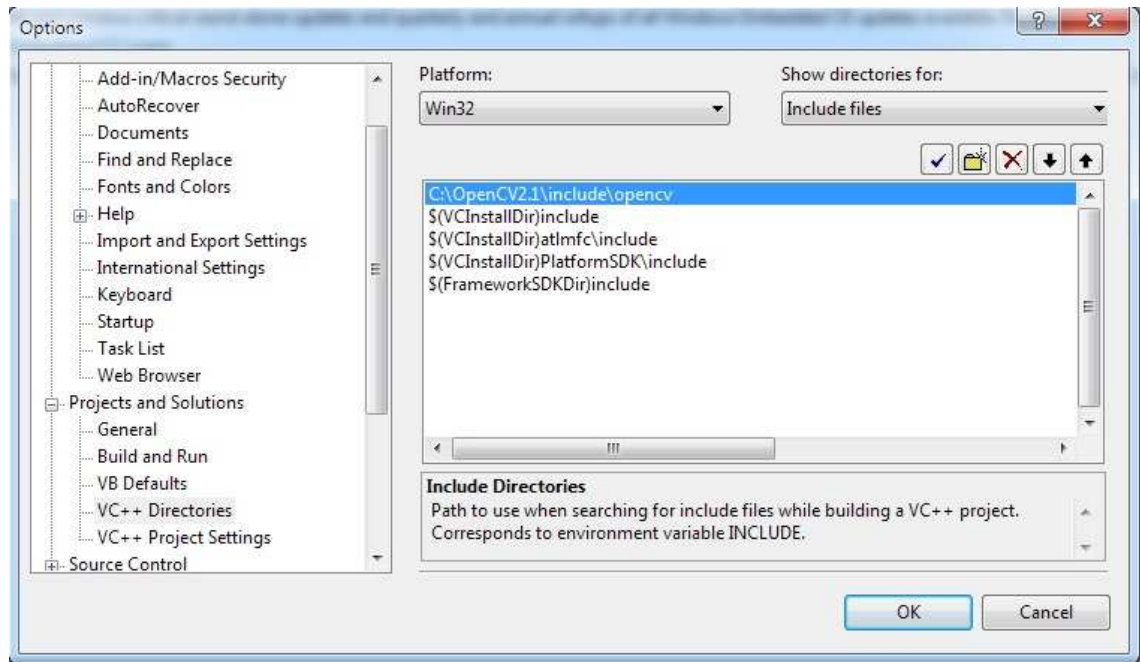
3.1.3 Microsoft Visual Studio 2005 üzerine OpenCV kurulumu

1) “<http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.1/OpenCV-2.1.0-win32-vs2008.exe/download>” adresinden OpenCV'nin Win32 versiyonu indirilip, “C:\OpenCV2.1” klasörüne kurulmuştur.

2) Visual Studio üzerinde Tools-> Options -> Projects and Solutions -> VC++ Directories bölümünde Show Directories kısmında;

- “Include files” seçilip “C:\OpenCV2.1\include\opencv” klasörü eklenmelidir.
- “Library files” seçilip “C:\OpenCV2.1\lib” klasörü eklenmelidir.
- “Source files” seçilip “C:\OpenCV2.1\src\cv”, “C:\OpenCV2.1\src\cvaux”, “C:\OpenCV2.1\src\cxcore” ve “C:\OpenCV2.1\src\highgui” klasörleri eklenmelidir (Şekil 3.5) [16].

3) Visual Studio içinde proje oluşturulduktan sonra “Project Properties” bölümünden sol menüde “Linker”->Input seçilerek “Additional Dependencies” kısmına “cv210.lib cxcore210.lib highgui210.lib” yazılarak derleme sırasında projenin OpenCV kütüphane dosyalarına erişimine izin verilmelidir. Bu her bir proje için tekrarlanmalıdır.



Şekil 3.3. OpenCV'nin Visual Studio üzerine kurulumu

3.2. YÖNTEM

Proje dahilinde, 2 farklı yöntem kullanılmıştır:

- Gerçek zamanlı (akan) görüntünün üzerinde işlem yaparak renk belirlenmesi
- Durdurulmuş görüntü üzerinde işlem yaparak renk belirlenmesi

Her 2 yöntemde de kalibrasyon ve ölçüm olmak üzere ikişer yazılım geliştirilmiştir. Bu yazılımlarda kameradan alınan RGB formatındaki görüntü, ortamdaki aydınlığın etkilerini minimize etmek amacıyla HSV renk uzayı formatına dönüştürülmüştür. V (aydınlık) değeri ihmal edilirken, H ve S değerleri işleme dahil edilmiştir. HSV renk uzayına dönüştürülen görüntü üzerinde gürültü azaltma amacıyla 15x15 boyutunda medyan filtresi uygulanmıştır.

Yazılıma başlangıç olarak Visual Studio açılıp File ->New-> Project seçildikten sonra “Win32 Console Application” seçilip proje ismi girilmelidir. Daha sonra gelen sayfada ise “Empty Project” seçilip proje boş olarak oluşturulmuştur. “Source files” bölümüne “Add Item” ile kaynak dosyası eklenmelidir. Yazılımlarda kullanmış olan OpenCV fonksiyonları Ek 1’de verilmiştir.

3.2.1. Gerçek Zamanlı Görüntü İşleme Yöntemi

Bu yöntemde çok yakın mesafeye odaklanabilen USB bağlantılı 1.3 mp ağız içi kamera kullanılmıştır (Şekil 3.4 ve 3.5). Kameranın algılayıcısının düşük kalite ve çözünürlüklü olması aydınlatma farklarından çok etkilenmesine sebep olduğundan kamera dışı mümkün olduğunca yakın mesafeden (yaklaşık 5 mm) tutulup yalnızca kendi led ışık kaynağının baskın olması amaçlanmıştır.



Şekil 3.4. USB ağız içi kamera



Şekil 3.5. Kameranın lens ve ledlerden oluşan uç kısmı

3.2.1.1 Kalibrasyon

Kalibrasyon yazılımında akan görüntü işlenerek önce RGB'den HSV uzayına dönüştürülüp medyan filtresi uygulanarak ton (hue) ve doygunluk (saturation) değerlerinin ortalaması ekrana yazdırılması amaçlanmıştır. Program çalıştırılarak Vita Klasik renk skalasından her bir renk için referans değerler alınmıştır. Bu değerler not edilerek daha sonra renk ölçümü yazılımında referans değer olarak kullanılmıştır. Kalibrasyon için tasarlanan algoritma Şekil 3.7'de görülmektedir. Kalibrasyon yazılımının kaynak kodunu içeren "kalibrasyon.cpp" dosyası Ek 2'dedir.

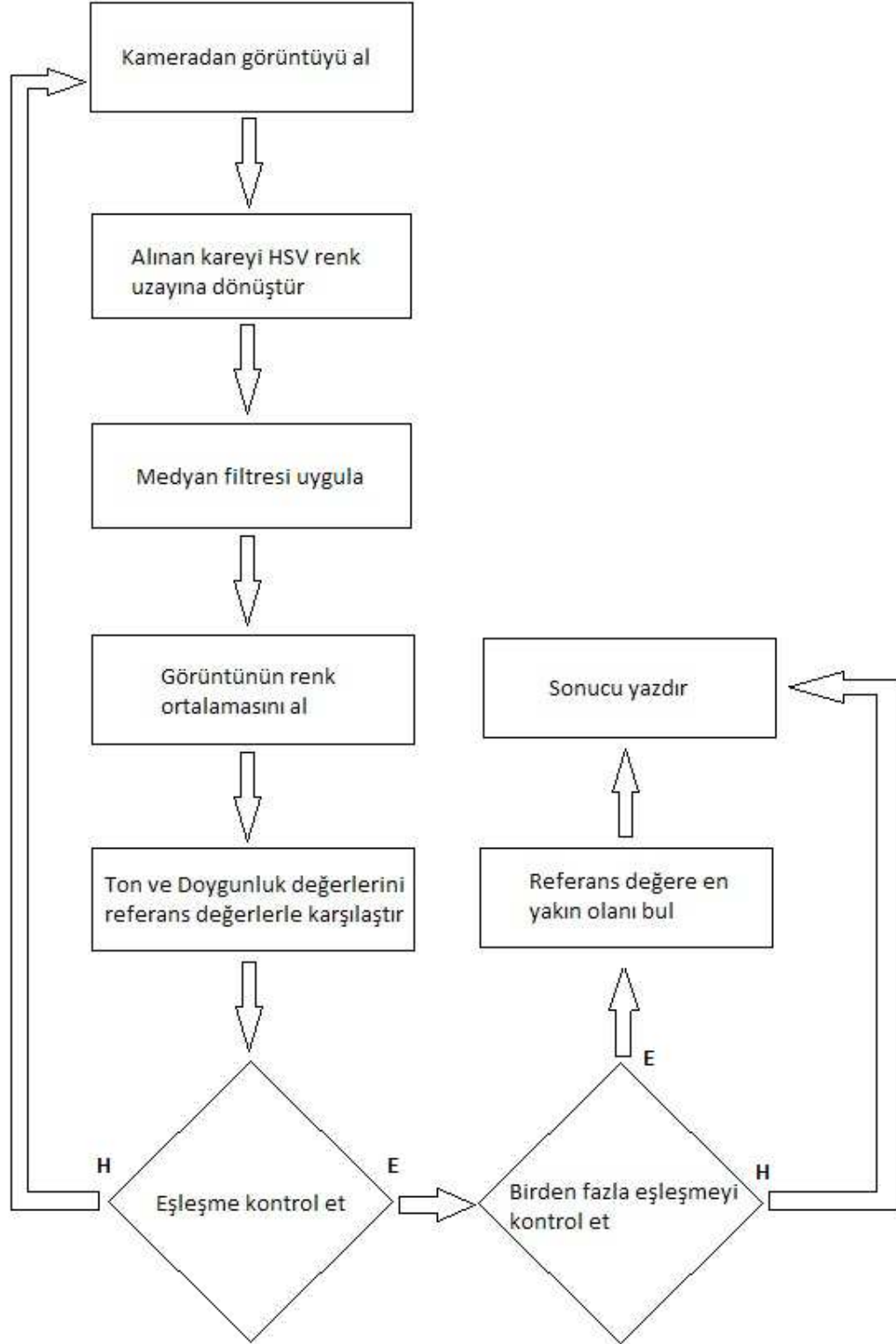


Şekil 3.6. Gerçek zamanlı görüntü işleme yöntemi için tasarlanan kalibrasyon yazılımının algoritması

3.2.1.2. Renk ölçümü

Bu yöntem için tasarlanan renk ölçümü yazılımında kalibrasyonda olduğu gibi akan görüntü HSV renk uzayı formatına dönüştürülmüş, üzerinde medyan filtresi uygulandıktan sonra alınan ton (hue) ve doygunluk (saturation) değerlerinin ortalamasının, kalibrasyon sırasında not edilen değerlerle karşılaştırılması amaçlanmıştır. Tasarlanan algoritmada (Şekil 3.7.) eğer birden fazla eşleşen değer varsa bunlardan referans değere en yakın olanının Vita klasik renk skalasına göre sonuç

olarak ekrana yazdırılması planlanmıştır. Renk ölçümü yazılımının kaynak kodunu içeren “renk.cpp” dosyası Ek 3’tedir.



Şekil 3.7. Gerçek zamanlı görüntü işleme yöntemi için tasarlanan renk ölçümü yazılımının algoritması

3.2.2. Durdurulmuş Görüntüyü İşleme Yöntemi

Bu yöntemde daha yüksek çözünürlüğe sahip USB bağlantılı 2 mp CMOS sensörlü, Carl Zeiss lense sahip Logitech Webcam Pro 9000 kullanılmıştır (Şekil 3.8). Kameranın minimum odaklanma mesafesi yeterince düşük olmadığından gerçek zamanlı görüntü işleme değil, görüntünün durdurulup yalnızca istenen bölgedeki rengin tespit edilmesi amaçlanmıştır. Ölçümler sırasında Logitech tarafından sağlanan kamera ayar programı ile odak mesafesi en yakına getirilmiştir. Yapılan denemeler sonucunda en yakın odak mesafesinin yaklaşık 22 mm olduğu tespit edilmiş ve ölçümlerde bu mesafe kullanılmıştır.



Şekil 3.8. Logitech Webcam Pro 9000

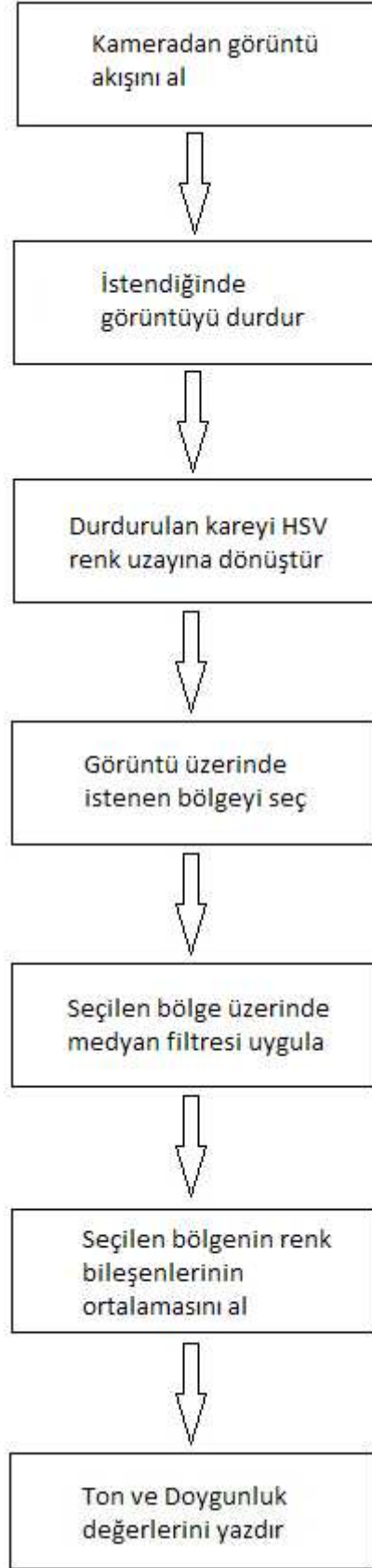
3.2.2.1. Kalibrasyon

Kullanılan kamera çok yakın mesafeye odaklanamadığından akıcı görüntü üzerinde değil, bir tuş yardımıyla durdurulmuş görüntü üzerinde işlem yapılarak ton (hue) ve doygunluk (saturation) değerlerinin ortalaması ekrana yazdırılması amaçlanmıştır. Durdurulan görüntüde istenen alan fare ile seçilmelidir. İlk yöntemde olduğu gibi, Vita klasik renk skalasından her bir renk için değerler alınıp not edilmiştir. Durdurulmuş

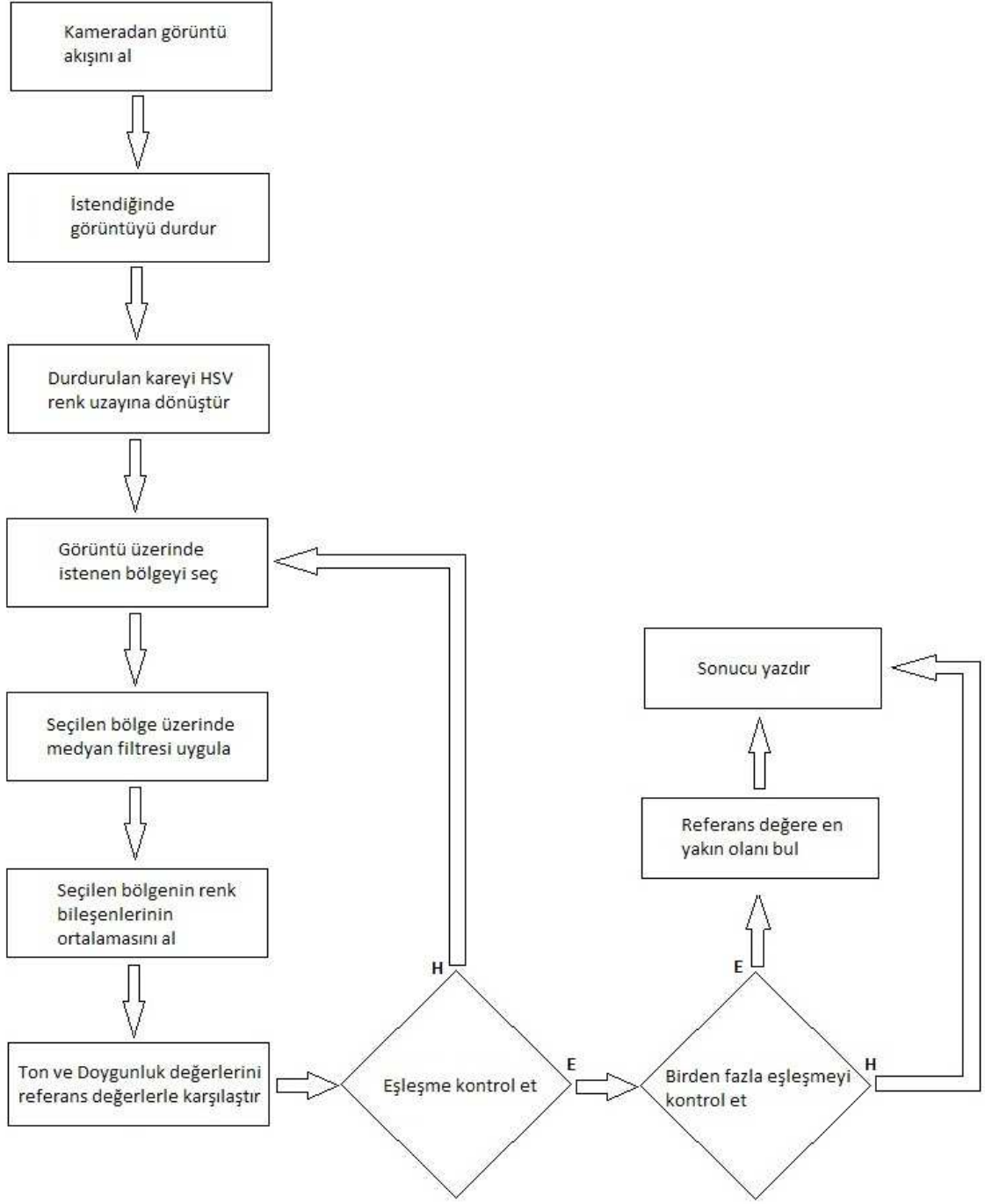
görüntü ile işlem yapılması amaçlanmış kalibrasyon yazılımının algoritması Şekil 3.9'daki gibidir. Bu yazılımın kaynak kodu ise Ek 4'tedir.

3.2.2.2. Renk ölçümü

Kalibrasyon aşamasında not edilen değerler yazılımda Bu yöntemdeki renk ölçümü yazılımında kalibrasyon yazılımı için düşünülen algoritma ton (hue) ve doygunluk (saturation) değerlerinin ortalaması alınana kadar aynı olup, sonrasında bu değerlerin kalibrasyonda alınan referans değerlerle karşılaştırılması amaçlanmıştır. Bu yöntemde de birden fazla eşleşen renk varsa referans değere en yakın olanının seçilmesi istenmiştir (Şekil 3.10). Yazılımın kaynak kodu Ek 5'tedir.



Şekil 3.9. Durdurulan görüntüyü işleme yöntemi için tasarlanmış kalibrasyon algoritması



Şekil 3.10. Durdurulan görüntüyü işleme yöntemi için tasarlanmış renk ölçümü algoritması

4. BULGULAR

4.1. Gerçek Zamanlı Görüntü İşleme Yöntemindeki Bulgular

Öncelikle kalibrasyon yazılımını çalıştırılarak Vita klasik renk skalasından kamera ile ton ve doygunluk değerleri alınmıştır. Daha sonra renk Ölçümü programı Yalova Devlet Hastanesi Ağız ve Diş Polikliniğinde restorasyon amaçlı başvuran hastalar üzerinde, kamera ile çekimler yapmak suretiyle denenmiştir. Kalibrasyon aşaması Şekil 4.1’de, ölçüm aşaması da Şekil 4.2’de görülmektedir.



Şekil 4.1. Gerçek zamanlı görüntü işleme yöntemindeki kalibrasyon aşaması



Şekil 4.2. Gerçek zamanlı görüntü işleme yöntemindeki renk ölçümü aşaması

Kalibrasyonda her bir renge karşılık elde edilen ton ve doygunluk değerleri Tablo 4.1’de görülmektedir.

Tablo 4.1. Gerçek zamanlı görüntü işleme yönteminde elde edilen kalibrasyon sonuçları

Vita Klasik renk değeri	Ton (Hue)	Doygunluk (Saturation)
A1	114	18
A2	95	7
A3	103	10
A3,5	79	12
A4	70	15
B1	93	8
B2	100	10
B3	85	9
B4	70	11
C1	112	16

C2	113	20
C3	100	13
C4	58	15
D2	117	21
D3	113	22
D4	91	11

Uygulama sonucunda hekimlerin tespit ettiği renklerle oluşturulan sistemin tespit ettiği renkler ise Tablo 4.2’dir. Yaklaşık %14,2’lik bir başarı oranı mevcuttur.

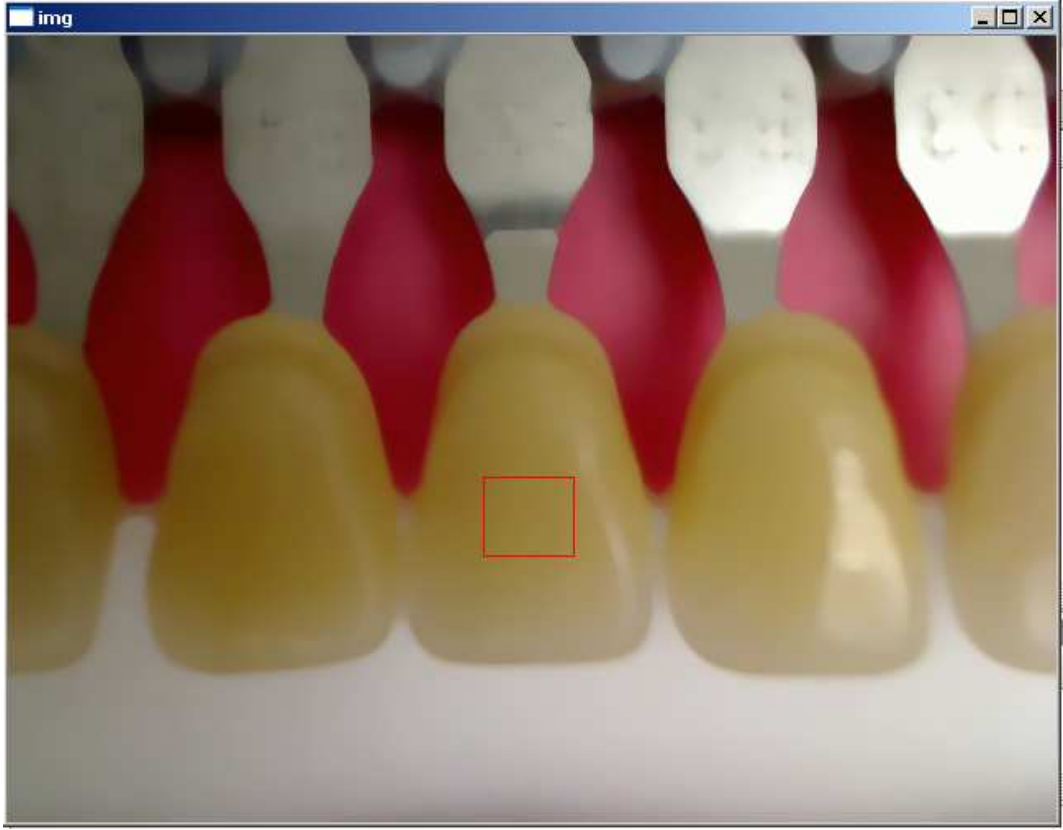
Tablo 4.2. Gerçek zamanlı görüntü işleme yönteminde elde edilen renk ölçümü sonuçları

Hasta	Hekim	Sistem
1	A3	C2
2	D2	C1
3	A3	C2
4	A2	B1
5	A1	A1
6	A2	B1
7	B2	B1

4.2. Durdurulmuş Görüntüyü İşleme Yönteminde Elde Edilen Bulgular

Kalibrasyon yazılımı çalıştırılıp, uygun çekim açısı bulunduğunda “enter” tuşuna basılarak görüntü durdurulmuştur. Rengi belirlenecek dişin yaklaşık orta kısmı (dentin) üzerinde fare ile bir alan seçilip ekranda beliren ton ve doygunluk değerleri not edilmiştir (Şekil 4.3 ve 4.4).

Bu yöntemdeki kalibrasyon yazılımında her bir renge karşılık elde edilen ton ve doygunluk değerleri Tablo 4.3’dir.



Şekil 4.3. Durdurulmuş görüntüyü işleme yöntemindeki kalibrasyon aşaması (1)



Şekil 4.4 Durdurulmuş görüntüyü işleme yöntemindeki kalibrasyon aşaması (2)

Tablo 4.3. Durdurulmuş görüntüyü işleme yönteminde elde edilen kalibrasyon sonuçları

Vita Klasik renk değeri	Ton (Hue)	Doygunluk (Saturation)
A1	21	87
A2	21	117
A3	21	131
A3,5	21	158
A4	21	176
B1	20	80
B2	22	109
B3	22	152
B4	22	171
C1	22	103
C2	22	135
C3	21	147
C4	20	201
D2	21	98
D3	21	123
D4	22	140

Elde edilen veriler renk ölçümü yazılımına girilip hastane ortamında deneyler yapılmıştır. Ulaşılan sonuçlarda Tablo 4.4'te de görüldüğü gibi %50 başarı oranı yakalanmıştır.

Tablo 4.4. Durdurulmuş görüntüyü işleme yönteminde elde edilen renk ölçüm sonuçları

Hasta	Hekim	Sistem
1	A2	A2
2	D2	A2
3	A3	A3
4	A1	B1
5	B1	B1
6	B2	B2
7	A3,5	B4
8	A3	D4

5. TARTIŞMA VE SONUÇ

Renk analizi ve seçimi, renk parametrelerinin objektif ve doğru değerlendirilmesini gerektirir. Geleneksel olarak skalalar yardımıyla görsel renk seçimi subjektif analizlere ve renk uyumsuzluklarına sebep olmaktadır. Elektronik renk değerlendirme sistemlerinin kullanılması ile subjektif renk analizinin doğurduğu dezavantajlar yok edilmektedir. Bu bağlamda geliştirilen spektrofotometre ve kolorimetreler çok yüksek maliyetleri ile birlikte, sınırlı hassasiyetleri ve kullanımlarındaki zorluklar nedeniyle yeterince ilgi görmemiştir. Bu yüzden alternatif elektronik sistemlere ihtiyaç duyulmuştur.

Bu tezde bilgisayara takılabilen kameralar kullanılarak gerçek zamanlı veya durdurulmuş görüntü işleme yaklaşımıyla diş rengi analizi konusu incelenmiştir. Yapılan çalışma ve deneyler sonunda, gerçek zamanlı görüntü işleme yönteminde kullanılan kameranın algılayıcısının düşük çözünürlüklü ve ışığa karşı çok hassas olması sebebiyle kameranın dişe çok yaklaştırılarak ölçüm yapılması gerektiği, aksi takdirde sistemin verimli çalışmadığı tespit edilmiştir. Kameranın dişe olan mesafesi, ortamdaki diğer ışık kaynakları gibi etkenler sonuçlarda belirgin değişimlere sebep olmaktadır. Sonuçta ancak % 14,2 oranında bir doğruluk yakalanabilmiştir.

Yüksek çözünürlüklü ve nispeten kaliteli lense sahip kameranın kullanıldığı diğer yöntemde ise tatmin edici sonuçlar alınmakla beraber, bu yöntemde de ortamdaki aydınlatma faktörü sonuçları etkilemektedir. Sabit ışık kaynağı altında ve dişe yaklaşık 22 mm mesafeden ölçüm yapıldığında doğru veya doğruya yakın sonuçlar elde edilebilmiştir. Ulaşılan başarı oranı %50'dir.

Yalnızca kamera ve günümüzde hali hazırda hemen her diş hekiminde bulunan bilgisayar gerektiren sistemi düşük maliyeti cazip kılmakla beraber, gelecek çalışmalarda daha iyi sonuçlar için CCD sensörlü, yüksek çözünürlüklü ve ortam ışığını bastırarak dahili aydınlatmaya sahip kameralar kullanılmalıdır. Ayrıca çekim mesafesini sabitlemek amacıyla uygun aparatların kullanımıyla daha tutarlı sonuçlara ulaşılması mümkündür.

KAYNAKLAR

1. BAYINDIR, F. ve WEE, G.A., 2006, Diş rengi seçiminde bilgisayar destekli sistemlerin kullanımı, Hacettepe Diş Hekimliği Fakültesi Dergisi, 30 (3), 40-46.
2. KEYF, F., UZUN, G., ALTUNSOY, S. , 2009, Diş hekimliğinde renk seçimi, Hacettepe Diş Hekimliği Fakültesi Dergisi, 33 (4), 52-58.
3. GÜREL, G., 2004, Porselen laminate venerler bilim ve sanatı, Quintessence yayınları, İstanbul, T-18-15.
4. SEGHI, R.R., JOHSTON, W.M. and O'BRIEN, W.J., 1989, Performance assesment of colorimetric devices on dental porcelains, J Dent Res, 68, 1755-9.
5. SEGHI, R.R, HEWLETT, E.R. and KIM, J., 1989, Visual and instrumental colorimetric assesment ofs of small color differences on traslucent dental porcelain, J Dent Res, 68:1760-4.
6. MILLER, L.L., 1993, Shade selection, J Esthet Dent, 4, 143
7. PRIEST, G and LINDKE, L, 2000, Tooth color selection and characterization accomplished with optical mapping, Pract Periodontics Aesthet Dent, 12(5):497-503.
8. PAUL, S., PETER, A., PIETROBON, N. and HAMMERLE, C.H., 2002, Visual and spectrophotometric shade analysis of human teeth, J Dent Res, 81 (8):578-82.
9. SIM, C.P., YAP, A.U. and TEO, J., 2001, Color perception among different dental personel, Oper Dent, 26(5):435-9.
10. PARAVINA, R.D., 2002, Evaluation of a newly developed visual shade-matching apparatus: Int J Prosthodont, 15(6):528-34.

11. ROBERT, D., 2000, *First contacts with the ShadeScan System*. Canadian J Dent Technol, 4, 52-60.
12. CHERKAS, L.A., 2001, *Communicating Precise Color Matching and Cosmetic Excellence*, Dentistry Today, 20 (4), 52-57
13. ÇAYIRLI, M., 2006, *Yanma Olayının Modellenmesi ve Görüntü İşleme Yoluyla Yanma Performansının Optimizasyonu*, Yüksek Lisans Tezi, T.C. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü
14. TOPRAK, A., GÜLER, İ., 2009, *Bulanık uyarlamalı ortalama filtresi kullanılarak MR görüntülerindeki darbe gürültüsünün azaltılması*, 5. Uluslararası İleri Teknolojiler Sempozyumu (IATS'09), Karabük, Türkiye
15. TAŞKIN, D., 2007, *Sıkıştırılmış video akımının düzensiz haritalar ve başlangıç kodlarına dayalı şifrelenmesi*, Doktora Tezi, T.C. Trakya Üniversitesi Fen Bilimleri Enstitüsü
16. DANACI, H.F., 2009, *OpenCV'nin kurulumu*, Visual C++ 2005 Express ve 2008 Express ile kullanımı, <http://hasanfehmi.wordpress.com/2009/06/11/opencv-nedir-visual-c-2005-ve-2008-express/>
17. 2010, *OpenCV2.1 C Reference*, <http://opencv.willowgarage.com/documentation/c/index.html>

EKLER

Ek 1. Yazılımda Kullanılan OpenCV Yapı ve Fonksiyonları

Yazılımda kullanılan fonksiyonlar ve tanımları şöyledir [17]:

Include Dosyaları

C++ yazılımlarında OpenCV kütüphanesindeki temel fonksiyonlar ve kullanıcı arabirimi fonksiyonlarını kullanabilmek amacıyla cv.h ve highgui.h include dosyaları dosyanın başına eklenmelidir.

```
#include "cv.h"  
#include "highgui.h"
```

IplImage

IplImage yapısı yeni görüntü işaretçisi deklare etmek amacıyla kullanılır.

```
typedef struct _IplImage  
{  
    int nSize;  
    int ID;  
    int nChannels;  
    int alphaChannel;  
    int depth;  
    char colorModel[4];  
    char channelSeq[4];  
    int dataOrder;  
    int origin;  
    int align;  
    int width;  
    int height;  
    struct _IplROI *roi;
```

```

struct _IplImage *maskROI;
void *imageId;
struct _IplTileInfo *tileInfo;
int imageSize;
char *imageData;
int widthStep;
int BorderMode[4];
int BorderConst[4];
char *imageDataOrigin;
}
IplImage;

```

CvScalar

CvScalar yapısıyla 4 elemanlı dizi tanımlanabilir.

```

typedef struct CvScalar
{
    double val[4];
}
CvScalar;

```

CvCapture

OpenCV kütüphanesinde sisteme bağlı video aygıtından görüntü almak için tanımlanan yapı “CvCapture” yapısıdır. Görüntü alınmaya başlanmadan önce bu yapı ile deklarasyon yapılmalıdır.

```

typedef struct CvCapture CvCapture()

```

cvCaptureFromCAM

Belirtilen bağlama noktasındaki kameradan görüntü alımını başlatır.

```

CvCapture* cvCaptureFromCAM(int index)

```

- index: Bilgisayara bağlı kameranın index numarası.

cvSetCaptureProperty

Kameradan görüntü alımıyla ilgili seçenekleri ayarlamak veya değiştirmek için kullanılır.

```
int cvSetCaptureProperty(CvCapture\* capture, int property_id, double value)
```

- capture: Görüntü yakalama işaretçisi
- property_id: Görüntü yakalama seçenekleri
- value: Değer

cvQueryFrame

Kameradan alınan görüntüyü belirtilen görüntü işaretçisine yükler.

```
IplImage\* cvQueryFrame(CvCapture\* capture)
```

- capture: Görüntü yakalama işaretçisi

cvReleaseCapture

Görüntü yakalama işlemi bittiğinde hafızayı temizlemek için kullanılır.

```
void cvReleaseCapture(CvCapture\*\* capture)
```

- capture: Görüntü yakalama işaretçisi

cvNamedWindow

Yeni bir pencere oluşturmak için kullanılır.

```
int cvNamedWindow(const char* name, int flags)
```

- name: Pencere ismi
- flags: Pencere parametreleri

cvDestroyAllWindows

Oluşturulan pencerelerin tümünü temizlemek için kullanılır.

```
void cvDestroyAllWindows(void)
```

cvShowImage

Belirtilen pencerede, istenen görüntünün gösterilmesi amacıyla kullanılır.

```
void cvShowImage(const char* name, const CvArr\* image)
```

- name: Görüntünün gösterileceği pencerenin ismi
- image: Gösterilecek görüntünün işaretçisi

cvReleaseImage

Belirtilen görüntüyle yapılacak iş kalmadığında hafızayı temizlemek için kullanılır.

```
void cvReleaseImage(IplImage\* image)
```

- image: Hedef görüntünün işaretçisi

cvInitFont

Görüntü üzerine yazdırılacak yazının tipini ayarlamak için kullanılır.

```
void cvInitFont(CvFont\* font, int fontFace, double hscale, double vscale, double shear=0, int thickness=1, int lineType=8)
```

- font: Yazı tipi işaretçisi
- fontFace: Yazı tipi
- hscale: Yatay ölçü
- vscale: Dikey ölçü
- shear: Yatıklık
- thickness: Kalınlık
- lineType: Çizgi formatı

cvCvtColor

Görüntüyü bir renk uzayından başka bir renk uzayına dönüştürmek için kullanılan fonksiyondur.

```
void cvCvtColor(const CvArr\* src, CvArr\* dst, int code)
```

- src: Kaynak dosya
- dst: Hedef dosya

code: Dönüştürme kuralı (CV_BGR2HSV, CV_BGR2GRAY vs.)

cvSetMouseCallback

Fare operasyonlarını yürüten fonksiyondur.

```
void cvSetMouseCallback(const char* windowName, CvMouseCallback onMouse,
void* param=NULL)
```

- *windowName*: Pencere ismi

- *onMouse*: Fare operasyonları:

- CV_EVENT_MOUSEMOVE: Fare hareket ettirilmesi
- CV_EVENT_LBUTTONDOWN: Sol tuşa basılması
- CV_EVENT_RBUTTONDOWN: Sağ tuşa basılması
- CV_EVENT_MBUTTONDOWN: Orta tuşa basılması
- CV_EVENT_LBUTTONUP: Sol tuşun bırakılması
- CV_EVENT_RBUTTONUP: Sağ tuşun bırakılması
- CV_EVENT_MBUTTONUP: Orta tuşun bırakılması
- CV_EVENT_LBUTTONDBLCLK: Sol tuşa çift tıklanması
- CV_EVENT_RBUTTONDBLCLK: Sağ tuşa çift tıklanması
- CV_EVENT_MBUTTONDBLCLK: Orta tuşa çift tıklanması

cvRect

Belirtilen koordinatlarda bir dikdörtgen alanı belirler.

```
CvRect cvRect( int x, int y, int width, int height );
```

- *x*: Dikdörtgenin sol üst köşe x koordinatı

- *y*: Dikdörtgenin sol üst köşe y koordinatı

- *width*: Genişlik

- *height*: Yükseklik

cvSetImageROI

Görüntü üzerinde yalnızca belirli bir alanda işlem yapılmak istendiğinde kullanılır. Bu fonksiyon çağırıldıktan sonraki görüntü işleme fonksiyonları belirlenen bölgeye uygulanır.

`void cvSetImageROI(IplImage* image, CvRect rect)`

- image: Hedef görüntünün işaretçisi
- rect: Önceden belirlenmiş dikdörtgen

cvResetImageROI

Görüntüde daha önce seçilen işlem yapılacak alanı sıfırlar.

- `void cvResetImageROI(IplImage* image)`
- image: Hedef görüntünün işaretçisi

cvSmooth

Gürültü azaltma amacıyla kullanılan yumuşatma filtresi fonksiyonudur.

`void cvSmooth(const CvArr* src, CvArr* dst, int smoothtype=CV_GAUSSIAN, int param1=3, int param2=0, double param3=0, double param4=0)`

- src: Kaynak dosya
- dst: Hedef dosya
- smoothtype: Yumuşatma kuralı (*CV_BLUR*, *CV_GAUSSIAN*, *CV_MEDIAN*)
- param1, param2, param3, param4: Yumuşatma kuralında belirtilen filtrenin alacağı parametreler.

cvAvg

Bir görüntüdeki her kanalın ortalamasını alan fonksiyon `cvAvg`'dir.

`CvScalar cvAvg(const CvArr* arr, const CvArr* mask=NULL)`

- arr: kaynak dizi
- mask: isteğe bağlı maske operatörü.

cvPutText

Görüntü üzerine yazı yazdırmak için kullanılır.

`void cvPutText(CvArr* img, const char* text, CvPoint org, const CvFont* font, CvScalar color)`

- img: Kaynak görüntü
- text: Kaynak metin
- org: Görüntü üzerindeki istenen koordinat (x,y)
- font: Yazı tipi işaretçisi
- color: Renk

cvWaitKey

Klavyeden herhangi bir tuşa basılmasını bekler. Basılan tuşun ASCII değerini döndürür.

```
int cvWaitKey(int delay=0)
```

- delay: Bekleme süresi

Ek 2. kalibrasyon.cpp Kaynak Kodu

```

////////////////////////////////////
// kalibrasyon.cpp
// Yazan: Aytaç Altan
// Referans H ve S değerlerinin alındığı yazılım
////////////////////////////////////

#include "cv.h"
#include "highgui.h"

CvCapture *capture = 0;
IplImage* img_hsv, * frame;
CvFont font;
CvScalar ortalama;
char sonuc_h[3], sonuc_s[3];

int enter, piksel_h, piksel_s;

int main()
{
    capture = cvCaptureFromCAM(0);
    cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_WIDTH, 640);
    cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT, 480);

    if (!capture){

        fprintf( stderr, "Camera takili degil!\n" );
        return 1;

    }

    frame = cvQueryFrame(capture);
    img_hsv=cvCreateImage(cvGetSize(frame),8,3);

```

```

cvInitFont(&font, CV_FONT_HERSHEY_SIMPLEX, .8, .8, 0, 2, 8);
cvNamedWindow("img", 1);

do{

    while(enter!=13)
    {

        frame = cvQueryFrame(capture);

        if(!frame) break;

        cvCvtColor(frame,img_hsv,CV_BGR2HSV);
        cvSmooth(frame,frame, CV_MEDIAN,15,15);

        ortalama = cvAvg(img_hsv);
        piksel_h = (int) ortalama.val[0];
        piksel_s = (int) ortalama.val[1];

        sprintf(sonuc_h, "%s%d", "Hue=", piksel_h);

        cvPutText(
            frame,
            sonuc_h,
            cvPoint(50,50),
            &font,
            CV_RGB(255, 0, 0)
        );

        sprintf(sonuc_s, "%s%d", "Sat=", piksel_s);

        cvPutText(
            frame,
            sonuc_s,
            cvPoint(50, 75),
            &font,
            CV_RGB(255, 0, 0)
        );

        cvShowImage("img", frame);

        enter = cvWaitKey(1);

        if(enter==27)
        {
            cvDestroyAllWindows();
            cvReleaseCapture(&capture);
            cvReleaseImage(&img_hsv);
            return (0);
        }

    }

}while(enter!=27);

/* Temizlik */

cvReleaseCapture(&capture);
cvDestroyAllWindows();
cvReleaseImage(&img_hsv);

```

```

    return (0);
}

```

Ek 3. renk.cpp Kaynak Kodu

```

////////////////////////////////////
// renk.cpp
// Yazan: Aytaç Altan
// Kalibrasyonla elde edilen H ve S değerleriyle karşılaştırma //
// yaparak sonucu gösteren yazılım
//
////////////////////////////////////

#include "cv.h"
#include "highgui.h"

CvCapture *capture = 0;
IplImage* img_hsv, * frame;
CvFont font;
CvScalar ortalama;
char sonuc[4];

int enter, piksel_h, piksel_s, i, j;
int eslesenler_h[16], eslesenler_s[16];

// Referans alınan değerlerden oluşturulan Look-up tablosu
char* renkler[16] = {"A1", "A2", "A3", "A3.5", "A4", "B1", "B2", "B3",
"B4", "C1", "C2", "C3", "C4", "D2", "D3", "D4"};
int renkler_h[16] = { 114, 95, 113, 70, 70, 93, 100, 79,
100, 113, 91, 70, 58, 117, 85, 103};
int renkler_s[16] = { 18, 7, 22, 11, 15, 8, 10, 12,
13, 20, 11, 15, 15, 21, 9, 10};

int main()
{
    capture = cvCaptureFromCAM(0);
    cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_WIDTH, 640);
    cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT, 480);

    if (!capture){

        fprintf( stderr, "Camera takili degil!\n" );
        return 1;

    }

    frame = cvQueryFrame(capture);
    img_hsv=cvCreateImage(cvGetSize(frame),8,3);
    cvInitFont(&font, CV_FONT_HERSHEY_SIMPLEX, .8, .8, 0, 2, 8);
    cvNamedWindow("img", 1);

    do{

        while(enter!=13)

```

```

{
    frame = cvQueryFrame(capture);

    if(!frame) break;

    cvCvtColor(frame, img_hsv, CV_BGR2HSV);
    cvSmooth(frame, frame, CV_MEDIAN, 15, 15);

    ortalama = cvAvg(img_hsv);
    piksel_h = (int) ortalama.val[0];
    piksel_s = (int) ortalama.val[1];

    for (i=0;i<16;i++){
        eslesenler_h[i]=0;
        eslesenler_s[i]=0;
    }

    for (i=0;i<16;i++)
    {
        if ( (piksel_h<renkler_h[i]+5 &&
piksel_h>renkler_h[i]-5) && (piksel_s<renkler_s[i]+5 &&
piksel_s>renkler_s[i]-5) )
        {

            eslesenler_h[i]=piksel_h;
            eslesenler_s[i]=piksel_s;
            sprintf(sonuc, "%s", renkler[i]);

        }else{

            if(i==15)
            {

                sprintf(sonuc, "%s",
"Hesaplaniyor");
            }
        }
    }

    for (i=0;i<16;i++)
    {
        for(j=0;j<16;j++)
        {
            if ( eslesenler_h[i]!=0 &&
eslesenler_s[i]!=0
                    && abs(piksel_h-
eslesenler_h[i])<abs(piksel_h-eslesenler_h[j])
                    && abs(piksel_s-
eslesenler_s[i])<abs(piksel_s-eslesenler_s[j]) )
            {
                sprintf(sonuc, "%s", renkler[i]);
            }
        }
    }

    cvPutText(
        frame,

```

```

        sonuc,
        cvPoint(50, 50),
        &font,
        CV_RGB(255, 0, 0)
    );

    cvShowImage("img", frame);

    enter = cvWaitKey(1);

    if(enter==27)
    {
        cvDestroyAllWindows();
        cvReleaseCapture(&capture);
        cvReleaseImage(&img_hsv);
        return (0);
    }
}

}while(enter!=27);

/* Temizlik */
cvReleaseCapture(&capture);
cvDestroyAllWindows();
cvReleaseImage(&img_hsv);

return (0);
}

```

Ek 4. kalibrasyon_roi.cpp Kaynak Kodu

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// kalibrasyon_roi.cpp //
// Yazan: Aytaç Altan //
// Durdurularak görüntü işleme yöntemi için tasarlanmış //
// kalibrasyon yazılım //
// //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include "cv.h"
#include "highgui.h"

CvCapture* capture;
IplImage* img_load, * img_rgb, * img_hsv, * frame;
CvFont font;
CvScalar ortalama;
CvPoint point2, baslangic_kord;
char sonuc_h[3], sonuc_s[3];
char* load;
int enter, drag, piksel_h, piksel_s, piksel_v;
uchar* piksel;

void mouseHandler(int event, int x, int y, int flags, void* param)
{

```

```

if (event == CV_EVENT_LBUTTONDOWN && !drag)
{
    baslangic_kord = cvPoint(x, y);
    drag = 1;
}

if (event == CV_EVENT_MOUSEMOVE && drag)
{
    img_rgb=cvCloneImage(img_load);

    cvRectangle(

    img_rgb,

    baslangic_kord,

    cvPoint(x, y),

    CV_RGB(255, 0, 0),

    1, 8, 0

    );

    cvShowImage("img", img_rgb);
}

if (event == CV_EVENT_LBUTTONUP && drag)
{

    img_rgb=cvCloneImage(img_load);
    point2 = cvPoint(x,y);

    if(point2.x==baslangic_kord.x &&
point2.y==baslangic_kord.y){

        // mouse sürüklenmemişse sadece tıklanan noktadaki
piksel bilgilerini al
        piksel = cvPtr2D(img_hsv, y, x, NULL);
        piksel_h =piksel[0];
        piksel_s =piksel[1];

    }else{

        cvSetImageROI(

            img_hsv,

            cvRect(

                baslangic_kord.x,

                baslangic_kord.y,

                (x-baslangic_kord.x),

```

```

        (y-baslangic_kord.y)
    )
);

ortalama = cvAvg(img_hsv);

piksel_h = (int) ortalama.val[0];
piksel_s = (int) ortalama.val[1];
//piksel_v = (int) ortalama.val[2];
}

sprintf(sonuc_h, "%s%d", "Hue=", piksel_h);

/* Sonucun yazdırılacağı dikdörtgeni çiz*/
cvRectangle(
    img_rgb,
    cvPoint(x-70, y-26),
    cvPoint(x + 50, y + 30),
    CV_RGB(255, 255, 0),
    CV_FILLED,
    8, 0
);

/* Sonucu dikdörtgenin içine yazdır */
cvPutText(
    img_rgb,
    sonuc_h,
    cvPoint(x-70, y),
    &font,
    CV_RGB(255, 0, 0)
);

sprintf(sonuc_s, "%s%d", "Sat=", piksel_s);

cvPutText(
    img_rgb,
    sonuc_s,
    cvPoint(x-70, y+25),
    &font,
    CV_RGB(255, 0, 0)
);

cvResetImageROI(img_hsv);

cvShowImage("img", img_rgb);

drag = 0;
cvReleaseImage(&img_rgb);
}

if (event == CV_EVENT_RBUTTONDOWN)
{
    cvShowImage("img", img_rgb);
    drag = 0;
}
}

```

```

}

int main()
{
    capture = cvCaptureFromCAM(0);
    cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_WIDTH, 640);
    cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT, 480);

    if (!capture) {
        fprintf( stderr, "Camera takili degil!\n" );
        return 1;
    }

    frame = cvQueryFrame(capture);
    img_hsv=cvCreateImage(cvGetSize(frame),8,3);
    cvNamedWindow("img", 1);

    do{
        while(enter!=13){
            frame = cvQueryFrame(capture);

            if(!frame) break;

            cvShowImage("img", frame);

            enter = cvWaitKey(1);

            if(enter==27){
                cvDestroyAllWindows();
                cvReleaseCapture(&capture);
                return (0);
            }
        }

        cvInitFont(&font, CV_FONT_HERSHEY_SIMPLEX, .8, .8, 0, 1,
8);

        img_load=cvCloneImage(frame);

        cvSmooth(img_load,img_load,CV_MEDIAN,15,15);

        cvCvtColor(img_load,img_hsv,CV_BGR2HSV);

        cvSetMouseCallback("img", mouseHandler, NULL);

        cvShowImage("img", frame);

        enter=cvWaitKey(0);

    }while(enter!=27);

    /* Temizlik */
    cvReleaseCapture(&capture);
    cvDestroyAllWindows();
    cvReleaseImage(&img_load);

```



```

        cvReleaseImage(&img_rgb);
        cvReleaseImage(&img_hsv);

    return (0);
}

```

Ek 5. renk_roi.cpp Kaynak Kodu

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// renk_roi.cpp                                                                    //
// Yazan: Aytaç Altan                                                            //
// Durdurularak görüntü işleme yöntemi için referans                          //
// HSV değerlerini kullanılarak renk ölçümünün                                  //
// yapıldığı yazılım                                                            //
//                                                                              //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include "cv.h"
#include "highgui.h"

CvCapture *capture = 0;
IplImage* img_load, * img_rgb, * img_hsv, * frame;
CvFont font;
CvScalar ortalama;
CvPoint baslangic_kord, point2;
char sonuc[3];
int enter, drag, piksel_h, piksel_s, i, j, genis;
uchar* piksel;

// Referans alınan değerlerden oluşturulan Look-up tablosu
char* renkler[16] = {"A1", "A2", "A3", "A3.5", "A4", "B1", "B2", "B3",
"B4", "C1", "C2", "C3", "C4", "D2", "D3", "D4"};
int renkler_h[16] = { 21, 21, 21, 21, 21, 20, 22, 22,
22, 22, 22, 21, 20, 21, 21, 22};
int renkler_s[16] = { 87, 117, 131, 158, 176, 80, 109, 152,
171, 103, 135, 147, 201, 98, 123, 140};

int eslesenler_h[16]= {0}, eslesenler_s[16]={0};

void mouseHandler(int event, int x, int y, int flags, void* param)
{
    if (event == CV_EVENT_LBUTTONDOWN && !drag)
    {
        baslangic_kord = cvPoint(x, y);
        drag = 1;
    }

    if (event == CV_EVENT_MOUSEMOVE && drag)
    {
        img_rgb=cvCloneImage(img_load);

        cvRectangle(

            img_rgb,

```

```

        baslangic_kord,
        cvPoint(x, y),
        CV_RGB(255, 0, 0),
        1, 8, 0
    );

    cvShowImage("img", img_rgb);
}

if (event == CV_EVENT_LBUTTONDOWN && drag)
{
    img_rgb=cvCloneImage(img_load);
    point2 = cvPoint(x,y);

    if(point2.x==baslangic_kord.x &&
point2.y==baslangic_kord.y){

        // mouse sürüklenmemişse sadece tıklanan noktadaki
        piksel bilgilerini al
        piksel = cvPtr2D(img_hsv, y, x, NULL);
        piksel_h =piksel[0];
        piksel_s =piksel[1];

    }else{

        cvSetImageROI(

            img_hsv,

            cvRect(

                baslangic_kord.x,

                baslangic_kord.y,

                (x-baslangic_kord.x),

                (y-baslangic_kord.y)

            )

        );

        ortalama = cvAvg(img_hsv);

        piksel_h = (int) ortalama.val[0];
        piksel_s = (int) ortalama.val[1];
    }

    for (i=0;i<16;i++){
        eslesenler_s[i]=0;
    }
}

```

```

genis=0;

for (i=0;i<16;i++)
{
    if ( (piksel_h==renkler_h[i] ) &&
(piksel_s<renkler_s[i]+5 && piksel_s>renkler_s[i]-5) )
    {
        eslesenler_s[i]=piksel_s;
    }
    else{
        if(i==15){
            sprintf(sonuc, "%s", "Yeniden
deneyiniz");
            genis=100;
            break;
        }
    }
}

for (i=0;i<16;i++)
{
    for(j=0;j<16;j++)
    {
        if ( eslesenler_s[i]!=0 && abs(piksel_s-
eslesenler_s[i])<abs(piksel_s-eslesenler_s[j]) )
        {
            sprintf(sonuc, "%s", renkler[i]);
        }
    }
}

/* Sonucun yazdırılacağı dikdörtgeni çiz*/
cvRectangle(
    img_rgb,
    cvPoint(x-70, y-26),
    cvPoint(x + 50 + genis, y+4),
    CV_RGB(255, 255, 0),
    CV_FILLED,
    8, 0
);

/* Sonucu dikdörtgenin içine yazdır */
cvPutText(
    img_rgb,
    sonuc,
    cvPoint(x-70, y),
    &font,
    CV_RGB(255, 0, 0)
);

cvResetImageROI(img_hsv);

cvShowImage("img", img_rgb);

```

```

        drag = 0;
        cvReleaseImage(&img_rgb);
    }

    if (event == CV_EVENT_RBUTTONDOWN)
    {
        cvShowImage("img", img_rgb);
        drag = 0;
    }
}

int main()
{
    capture = cvCaptureFromCAM(0);
    cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_WIDTH, 640);
    cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT, 480);

    if (!capture) {
        fprintf( stderr, "Camera takili degil!\n" );
        return 1;
    }

    frame = cvQueryFrame(capture);
    img_hsv=cvCreateImage(cvGetSize(frame),8,3);
    cvNamedWindow("img", 1);

    do{

        while(enter!=13){

            frame = cvQueryFrame(capture);

            if(!frame) break;

            cvShowImage("img", frame);

            enter = cvWaitKey(1);

            if(enter==27){
                cvDestroyAllWindows();
                cvReleaseCapture(&capture);
                return (0);
            }
        }

        cvInitFont(&font, CV_FONT_HERSHEY_SIMPLEX, .8, .8, 0, 1,
8);

        img_load=cvCloneImage( frame);

        cvSmooth(img_load,img_load,CV_MEDIAN,15,15);

        cvCvtColor(img_load,img_hsv,CV_BGR2HSV);

        cvSetMouseCallback("img", mouseHandler, NULL);

```

```
        cvShowImage("img", frame);

        enter=cvWaitKey(0);

    }while(enter!=27);

    /* Temizlik */
    cvReleaseCapture(&capture);
    cvDestroyAllWindows();
    cvReleaseImage(&img_load);
    cvReleaseImage(&img_rgb);
    cvReleaseImage(&img_hsv);

return (0);
}
```

ÖZGEÇMİŞ

Adı Soyadı : Aytaç ALTAN

Doğum Yeri : Yalova

Doğum Yılı : 1984

Medeni Hali : Bekâr

Eğitim Durumu:

Lise : Yalova Lisesi, 1998-2002

Lisans : Dumlupınar Üniversitesi, Elektrik-Elektronik Mühendisliği, 2002-2007

Yabancı Dil : İngilizce