

755782

**İŞİN İZLEME VE İŞİN ETKİNLİK YÖNTEMLERİYLE
AYDINLATMANIN BİLGİSAYAR ORTAMINDA BENZETİMİ**

İsmail Serkan ÜNCÜ

**DOKTORA TEZİ
(ELEKTRİK EĞİTİMİ)**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Ağustos 2004

ANKARA

İsmail Serkan ÜNCÜ tarafından hazırlanan İŞİN İZLEME VE İŞİN ETKİNLİK YÖNTEMLERİYLE AYDINLATMANIN BİLGİSAYAR ORTAMINDA BENZETİMİ adlı bu tezin Doktora tezi olarak uygun olduğunu onaylarım.



Doç. Dr. Osman GÜRDAL

Tez Yöneticisi

Bu çalışma, jürimiz tarafından Elektrik Eğitimi Anabilim Dalında Doktora tezi olarak kabul edilmiştir.

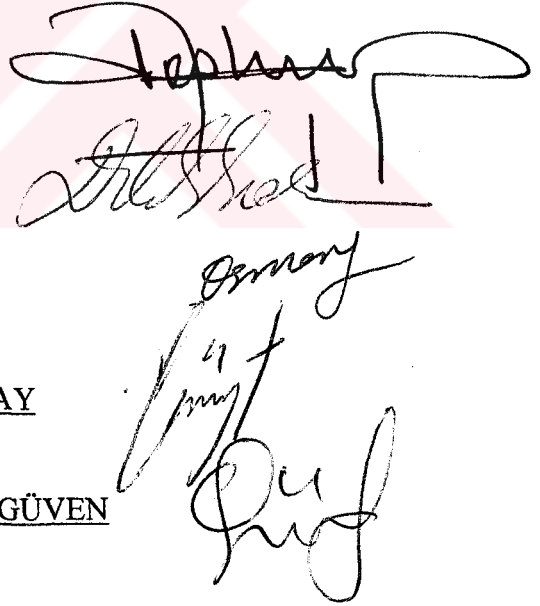
Başkan :Prof. Dr. İsmail COŞKUN

Üye :Doç. Dr. Dilek ENARUN

Üye :Doç. Dr. Osman GÜRDAL

Üye :Yard. Doç. Dr. Cüneyt KURTAY

Üye :Yard. Doç. Dr. Mehmet Emin GÜVEN



Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.



İÇİNDEKİLER

Sayfa

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
ÇİZELGELERİN LİSTESİ.....	iv
ŞEKİLLERİN LİSTESİ.....	v
EKLERİN LİSTESİ.....	xi
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ.....	1
2. IŞIK ŞİDDET EĞRİLERİYLE İLGİLİ ÇALIŞMALAR.....	5
2.1. Işın İzleme ve Işın Etkinlik Metotlarının Gelişimi.....	5
2.2. Aydınlatma Tasarımında Kullanılan Programlar.....	7
2.3. Işık Şiddet Eğrileri İle İlgili Tanımlamalar.....	9
2.3.1. Işık akısı.....	9
2.3.2. Işık şiddeti.....	10
2.4. Işık Dağılım Küresi.....	11
2.5. Işık Şiddeti Eğrisi.....	12
2.5.1 Bir ışık şiddeti eğrisi ile temsil edilebilen armatür.....	12
2.5.2 İki ışık şiddeti eğrisi ile temsil edilebilen armatür.....	13
2.5.3 Üç ve daha fazla ışık şiddeti eğrisi ile temsil edilebilen armatür.....	15
2.6. Işık Dağılım Yüzeyi.....	15
2.7. Fotometrik Yasalar.....	15

2.7.1 Kosinüs yasası	15
2.7.2 Lambert yasası.....	15
2.7.3 Uzaklıklar karesi ile ters orantı yasası.....	16
3. IŞIN İZLEME VE IŞIN ETKİNLİK METOTLARININ AYDINLATMA BENZETİMİNE UYGULANMASI	19
3.1. Işın İzleme Metodu.....	20
3.1.1. Işın izleme’de sadeleştirme teknikleri.....	21
3.1.2. Işın izleme ile ilgili tanımlamalar.....	21
3.1.3. Işın etkinlik ile görüntü oluşturma	24
3.2. Işın Etkinlik Metodu.....	28
3.2.1. Işın etkinlik ile görüntü oluşturma	29
3.2.2. Işın etkinlik’ te sadeleştirme teknikleri	30
3.2.3. Temel fotometrik tanımlamalar.....	31
3.3. Kaplama Eşitliği	34
3.3.1. Yerel veya doğrudan aydınlatma.....	34
3.3.2. Global veya dolaylı aydınlatma.....	35
3.3.3. Işın etkinlik eşitliği.....	38
3.3.4. Form faktörü.....	39
3.3.5. Sabit elemanların ışın etkinliği.....	40
3.3.6. Ağ oluşturma	42
4. BİLGİSAYAR KONTROLLÜ GONİOFOTOMETRE TASARIMI.....	44
4.1. Goniofotometre	45
4.1.1. Ölçüm düzlemi	46
4.1.2. Yapım prensipleri	48

4.2. Goniofotometre Seçimi	50
4.3. Motor Seçimi	50
4.3.1. Adım motorlarının uyartımı	51
4.4. Adım Motoru Sürücü Sistemi.....	53
4.5. Bilgisayar Ortamından Dış Ortama Bilgi Aktarımı	55
4.6. Mekanik Sistem.....	57
4.6.1. Sistem içinde motorların çalışması	57
4.6. Işık Şiddet Eğrilerin Elde Edilmesi	59
4.7. Sistemin Özellikleri	61
4.7.1. Sistemin hata kaynakları	61
5.GONİOFOTOMETRİK ÖLÇÜM YAZILIMININ TANITILMASI.....	62
5.1. Program Kontrol Alanı	62
5.1.1. Gonio konsolu	63
5.1.2. Veri havuzu konsolu.....	68
5.2. Işık Şiddeti Verilerinden Eğrilerinin Matematiksel Olarak Tanımlanması..	69
5.2.1. En küçük kareler metodu.....	69
5.2.2. En küçük kareler metodu ile verilen noktalardan geçen polinom denklemini bulunması	70
5.2.3. Modelleyici alanı.....	73
5.3. Işık Şiddeti Verilerinden Eğri ve Üç Boyutlu Görüntü Oluşturma	73
5.3.1. Işık şiddet eğrisinin elde edilmesi	74
5.3.2. En küçük kareler metodu yöntemiyle elde edilen eğrilerden üç boyutlu görüntü oluşturma	75
5.3.3. Üç boyutlu uygulama alanları	81

5.4 Yazılımın Kaynak Kodları	88
6. DENEYSEL SONUÇLAR	90
6.1. Işın İzleme Yöntemiyle Duvar Renklerinin Karşılaştırılması.....	95
6.2. Işın İzleme ve Işın Etkinlik Metotlarının Karşılaştırılması	96
7. SONUÇ	105
KAYNAKLAR	109
EKLER.....	113
ÖZGEÇMİŞ	150



İŞİN İZLEME VE İŞİN ETKİNLİK YÖNTEMLERİYLE AYDINLATMANIN BİLGİSAYAR ORTAMINDA BENZETİMİ

(Doktora Tezi)

İsmail Serkan ÜNCÜ

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Ağustos 2004

ÖZET

Bu çalışmada, bir armatürün ışık şiddet eğrilerinden üç boyutlu matematiksel model elde eden bir goniofotometre sistemi geliştirilmiştir. Bu model üç boyutlu tasarım alanlarında kullanılabilen bir yazılımla görsel hale getirilmiştir. Goniofotometrenin tüm fonksiyonları bilgisayar yardımıyla kontrol edilmektedir. Sistemde armatür adım motorları ile yatay ve dikey eksenlerde hareket ettirilmektedir. Adım motorları kontrolü bilgisayarın paralel portundan gönderilen sinyallerle sağlanmaktadır. Armatür hareket halindeyken lüks metre tarafından ölçülen değerler seri porttan bilgisayara aktararak saklanmaktadır. Işık şiddet eğrileri yazılımdaki yönlendirme ve hesaplamalarla elde edilmektedir. Her armatüre ait ışık şiddet eğrilerinden matematiksel model çıkartılmaktadır. Bulunan matematiksel modeller yazılımla oluşturulan bilgisayar ortamındaki üç boyutlu bir mekana ışın izleme ve ışın etkinlik yöntemleriyle ayrı ayrı aktararak armatürleri tasarım ekranlarında uygulama imkanı sağlanmaktadır. Ayrıca yazılıma matematiksel model denklemleri dışarıdan girilerek denkleme göre armatür tasarlama gibi işlevleri de üzerinde bulunmaktadır.

Bilim Kodu : 704
Anahtar Kelimeler : Işık şiddeti, Armatürler, Işın izleme, Işın etkinlik.
Sayfa Adedi : 150
Tez Yöneticisi : Doç. Dr. Osman GÜRDAL

**SIMULATION OF LIGHTING BY RAY TRACING AND RADIOSITY IN
COMPUTER ENVIROMENT**

(Ph.D. Thesis)

İsmail Serkan ÜNCÜ

**GAZI UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

July 2004

ABSTRACT

In this study, a goniophotometer system to obtain mathematical model luminous intensity curves of the luminaire is implemented. This model is developed visually by using three dimensional design software. All functions of the goniophotometer is controlled by computer. In the system, the luminaries are moved vertically and horizontally. All measured data obtained by a luxmeter is sent to computer by serial port and then saved as a file. Finally luminous intensity curves are obtained according to the instructions and calculations of the software. Each model of luminaries is obtained by the software program. Developed models provided are transferred to 3D enviroment of the computer program via ray tracing and radiosity individually. As a result, the design of luminaire is shown on the computer screen for applibility. Additionally, the external mathematical model can be applied to the software for luminaire design.

Science Code : 704
Key Words : Luminous intensity, Luminaire, Ray tracing, Radiosity
Page Number : 150
Adviser : Assoc. Prof. Dr. Osman GÜRDAL

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren değerli danışmanım Doç. Dr. Osman Gürdal' a ayrıca aydınlatma konusunda bilgimin ve ufkumun bu seviyeye gelmesinde çok yardımını gördüğüm Doç. Dr. Dilek Enarun'a, matematiksel modelin çıkartılmasındaki yardımlarından dolayı Yard. Doç. Dr. H. Hüseyin Sayan'a ve tüm mesai arkadaşlarıma teşekkürü bir borç bilirim.

Ayrıca manevi desteklerini her zaman yanımda hissettiğim sevgili Eşim ve Aileme teşekkür ederim.



ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. Tek-faz uyartımın faz uyartım sıralaması.....	51
Çizelge 4.2. İki-faz uyartımın faz uyartım sıralaması.....	52
Çizelge 4.3. Yarım adım (karma) uyartımın faz uyartım sıralaması.....	52
Çizelge 4.4 İleri yön için faz sıralaması (CCW).....	55
Çizelge 4.5 Geri yön için faz sıralaması (CCW).....	55
Çizelge 5.1. Açılara göre ışık şiddet verilerinin gösterilmesi	71
Çizelge 5.2 C katsayılarının gösterilmesi.....	72



ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Düzlemsel açıya ait uzay açının birim küre üzerinde gösterilmesi.....	10
Şekil 2.2 Bir ışık şiddeti eğrisi ile temsil edilebilen bir ışık kaynağının α doğrultusundaki I_α ışık şiddetinin tanımlanması	11
Şekil 2.3. Işık dağılım küresi.....	12
Şekil 2.4. Bir ışık şiddeti eğrisi ile temsil edilebilen armatürün ışık şiddet eğrisi ..	13
Şekil 2.5. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün ışık şiddet eğrileri	13
Şekil 2.6. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün enine kesiti.....	14
Şekil 2.7. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün boyuna kesiti	14
Şekil 2.8. Uzaklıklar karesi ile ters orantı yasası prensip şeması	16
Şekil 2.9 Uzay açısı izdüşümü yasası prensip şeması	18
Şekil 3.1. İğne deliği kamerası.....	22
Şekil 3.2. Göz, görüntü penceresi ve nesneler	23
Şekil 3.3. Işık kaynağından çıkan ışınların göze doğru yansıması	24
Şekil 3.4. Gölge oluşması	25
Şekil 3.5. Işın izlemede tam gölge oluşturma	25
Şekil 3.6. Işın izlemede yarım gölge oluşturma.....	26
Şekil 3.7. Işın izleme ile yarı saydam bir nesne oluşturma.....	26
Şekil 3.8. Kalın bir camın yoğunluğuna göre kırılma indisi özelliği.....	27
Şekil 3.9. Küre ve İki ışık şiddeti eğrisi ile temsil edilebilenler prizmasının aynadaki yansıması.....	28
Şekil 3.10. Işın etkinlik metodu kullanarak aydınlatması hesaplanmış bir oda.....	30
Şekil 3.11. Birim yüzeyi geçen ışınlar	31

Şekil 3.12. Küresel koordinat sisteminde diferansiyel alanın xy düzlemindeki izdüşümünün alınması.....	32
Şekil 3.13. İki nokta transfer geometrisi	37
Şekil 3.14. Üç nokta transfer geometrisi.....	38
Şekil 3.15. İki taban fonksiyonunun form faktörü	40
Şekil 3.16. İlk düğüm algoritmaları ile bölmeleme	43
Şekil 3.17. Düzgün geometrik şekillerin düğüm noktalarının belirlenmesi.....	43
Şekil 4.1. Sistemin çalışma yapısı.....	45
Şekil 4.2. A düzlemleri.....	46
Şekil 4.3. B düzlemleri.....	47
Şekil 4.4. Işık kaynağına dik bağlantılı C düzlemleri	48
Şekil 4.5. Işık kaynağını döndürme düzeneği olan goniofotometre Tip 1	48
Şekil 4.6. Işık kaynağını döndürme düzeneği olan goniofotometre Tip 2	49
Şekil 4.7. Işık kaynağını döndürme düzeneği olan goniofotometre Tip 3	49
Şekil 4.8. Adım motoru güç devresi.....	53
Şekil 4.9. Adım motorlarını çalıştıran sürücü devre	54
Şekil 4.10. Bilgisayar ara birim bağlantıları	56
Şekil 4.11. Birinci adım motorunun sisteme yaptırdığı hareket.....	57
Şekil 4.12. Işık kaynağının C düzleminde hareketi.....	58
Şekil 4.13. Sisteme bağlı İki ışık şiddeti eğrisi ile temsil edilebilen armatür şekli..	58
Şekil 4.14. Kurulan goniofotometre sisteminin fotoğrafı ve karanlık oda.....	60
Şekil 5.1. Localizer alanı.....	62
Şekil 5.2. Dosya, dil ve yardım menüleri.....	62
Şekil 5.3. Klasörden sisteme dosya ilave edilmesi	63

Şekil 5.4. Gonio konsolu.....	64
Şekil 5.5. Armatür tipi seçimi	65
Şekil 5.6. Yatay ölçüm sayısı ve dikey ölçüm sayısı alanları	65
Şekil 5.7. Veri alıcısı.....	66
Şekil 5.8. Veri alıcısı ve platform alanları	67
Şekil 5.9. Platform ayarları alanı seri ve paralel port seçimi	67
Şekil 5.10. Veri havuzu konsolu	68
Şekil 5.11. İnterpolasyon eğrilerinin gösterilmesi	70
Şekil 5.12. Denklem sayısı ve derecesi belirleme.....	73
Şekil 5.13. Işık şiddet vektörlerinin abak üzerine aktarılması	74
Şekil 5.14. İki boyutlu ışık şiddet eğrisi.....	75
Şekil 5.15. İki boyutlu ışık şiddet eğrilerinden üç boyutlu hacimlere geçiş	75
Şekil 5.16. (a)'da A noktası 0° derecede ve (b)' de bir A noktasının x eksenini etrafındaki 90° derece döndürülmesi ile belirlenen koordinatları.	76
Şekil 5.17. Üç boyutlu ışık dağılım yüzeyi	78
Şekil 5.18. Üç boyutlu ışık dağılım yüzeyinin kendi etrafında döndürülerek incelenmesi.....	78
Şekil 5.19. Üç boyutlu ışık dağılım yüzeyi renkli kafes biçimli gösterilmesi	79
Şekil 5.20. Üç boyutlu ışık dağılım yüzeyi siyah renkli kafes biçimli gösterilmesi	80
Şekil 5.21. Üç boyutlu ışık dağılım yüzeyinin katı biçimli gösterilmesi	80
Şekil 5.22. Işık şiddet eğrilerinin gerçek oda içerisine dağılımının gösterilmesi	81
Şekil 5.23. Gonio konsolundan ışın izleme ve ışın etkinlik alanlarına geçiş.....	81
Şekil 5.24. Üç boyutlu ortamda X, Y ve Z eksenlerinin gösterilmesi	82
Şekil 5.25. Örnek alan ve ayarlar penceresi	82

Şekil 5.26. Işık kaynağı koordinatlarındaki değişikliklerin örnek alan üzerine etkileri	83
Şekil 5.27. Kamera koordinatlarındaki değişikliklerin örnekteki alana etkisi	84
Şekil 5.28. Dosyadan üç boyutlu ortam yükleme	85
Şekil 5.29. 3D studio max programıyla oluşturulan örnek alanda ışın izleme yöntemiyle görüntü oluşturulması	86
Şekil 5.30. Işın etkinlik alanı başlangıç konumu	87
Şekil 5.31. Işın etkinliğe 1607 yansımadan sonra oluşan ışık dağılımı	88
Şekil 5.32. Projeyi oluşturan alt program klasörleri	89
Şekil 6.1. Bir ışık şiddeti eğrisi ile temsil edilebilen armatür	90
Şekil 6.2. Bir ışık şiddeti eğrisi ile temsil edilebilen armatürün goniofotometrik ölçümlerini yapan sistem	90
Şekil 6.3. Bir ışık şiddeti eğrisi ile temsil edilebilen armatürün 25° derecedeki ışık şiddet eğrisi	91
Şekil 6.4. Bir ışık şiddeti eğrisi ile temsil edilebilen armatürün 79° derecedeki ışık şiddet eğrisi	91
Şekil 6.5. Bir ışık şiddeti eğrisi ile temsil edilebilen armatürün 0° derecedeki ışık dağılım yüzeyi	91
Şekil 6.6. İki ışık şiddeti eğrisi ile temsil edilebilen armatür	92
Şekil 6.7. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün 0° derecedeki ışık şiddet eğrisi	93
Şekil 6.8. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün 47° derecedeki ışık şiddet eğrisi	93
Şekil 6.9 İki ışık şiddeti eğrisi ile temsil edilebilen armatürün 90° derecedeki ışık şiddet eğrisi	93
Şekil 6.10. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün 56° derecedeki ışık dağılım yüzeyi	94
Şekil 6.11. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün 93° derecedeki ışık dağılım yüzeyi	94

Şekil 6.12. (a) ve (b)'de aynı lamba farklı reflektör, (c)'de ise (a)'daki reflektörle aynı güçlü lamba benzetimi olmak üzere üç değişik armatürün ışık şiddet verilerini kullanarak yapılan ışın izleme karşılaştırılması.....	95
Şekil 6.13. Karşılaştırma için tavana yakın kısımlar büyütülmüştür.	96
Şekil 6.14. Işın izleme ve ışın etkinlik metotlarının 73 yansımadan sonra oluşan ışık dağılımı.....	97
Şekil 6.15. Işın izleme ve ışın etkinlik metotlarının 875 yansımadan sonra oluşan ışık dağılımı.....	97
Şekil 6.16. Işın izleme ve ışın etkinlik metotlarının 2801 yansımadan sonra oluşan ışık dağılımı.....	98
Şekil 6.17. Işın izleme ve ışın etkinlik metotlarının 3132 yansımadan sonra oluşan ışık dağılımı.....	98
Şekil 6.18. 3d max studio programında ışın izleme algoritmasının çalıştırılması. ..	99
Şekil 6.19. 3d max studio programında ışın etkinlik algoritmasının çalıştırılması	100
Şekil 6.20 Geliştirilen yazılımla 1607 yansıma sonunda elde edilen ışın etkinlik uygulaması.	101
Şekil 6.21. 3d max studio programıyla yapılan ışın etkinlik uygulaması	101
Şekil 6.22. Geliştirilen yazılımla 580 yansıma sonucunda elde edilen ışın izleme uygulaması	103
Şekil 6.23. Geliştirilen yazılımla 1607 yansıma sonucunda elde edilen ışın izleme uygulaması	103
Şekil 6.24. 3d studio max programıyla yapılan ışın izleme uygulaması.....	104

EKLERİN LİSTESİ**Ek****Sayfa**

Ek-1. Yazılımı oluşturan alt pencerelerin kodları..... 129



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simge ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklama

Φ	Işık akısı (Lümen)
ε_2	Geliş açısı (Derece)
I_α	α doğrultusundaki ışık şiddet değeri (Candela)
Ω_α	Uzay açısı (Steradyan)
dA	Kapalı alan içindeki birim alanı
dt	Bir foton havada kalış süresi (Saniye)
ω	Parçacıkların yönü,
L	Parıltı (Nit)
h	Planck sabiti
c	Işık hızı (m/sn ²)
E_{ni}	ΔS_i düzleminin aydınlık düzeyi (Lüks)
E_0	Birim alana dik olarak aydınlık düzeyi
R_0	Çıkan ışıntı değeri (Phot)
$G(x, x')$	Geometrik fonksiyonun değeri
$I(x, x')$	İki nokta arası aktarma yoğunluğu
R_e	Yutulan ışıntının değeri
B	Işın etkinlik
r	Yarıçap (Metre)
$p(x)$	Birim hacim başına ışın miktarı
N_i	Sabit taban fonksiyonlar
δ_{ij}	Kronecker delta
F_{ij}	Form faktörü
S	Yüzey alanı (m ²)

$E(x)$	Yutma fonksiyonu
ω	Işın yönü
v	Işın hızı (m/sn ²)
$\rho(x)$	Yayınım yansıması
ρ_i	i elemanın x noktasındaki yayınım yansıması
$\varepsilon(x)$	Hata fonksiyonu
$\tau(x)$	Rezidu fonksiyonu
K	İç hareket katsayıları matrisi
F	Form faktör matrisi
N_i	i.nci taban fonksiyonu
f	Çok yönlü yansıma fonksiyonu

Kısaltmalar

Açıklamalar

IES	Illumination Enginneering Society
CIE	International Commission on Illumination
ASE	ASCII Scene Export
CRT	Cathode Rate Tube
NURB	Non- Uniform Rational B-spline
BRDF	Bidirectional Reflection Distribution Function
CAD	Computer Aield Design
RS	Recommendand Standart

1. GİRİŞ

Işığın üretimi, dağıtımı, uygulamaları ve ölçülmesi aydınlatmanın temel konularıdır. Aydınlatmada amaç, ışığı isteğe yönelik olarak kullanmaktır. Kaynaktan çıkan ışık, armatür yardımıyla istenen şekle getirilebilir. Armatürler, lamba ya da lambaların ışık dağılımını düzenlemek, süzmek veya değiştirmek için kullanılırlar. Uygun armatürlerin uygun yerlerde kullanılması ile en önemli aydınlatma uygulaması gerçekleştirilmiş olur.

Doğru bir aydınlatma ile göz sağlığı korunur, görme yeteneği artar, kazalar azalır, yapılan işin verimi yükselir ve güvenlik sağlanır. Tüm dekorasyon düzenlerinde aydınlatılan nesnede oluşturulan görüntü nesneye anlam ve değer katar. Günümüzde bir odanın aydınlatılması için tavanının ortasına bir lamba asılması yeterli değildir. Aydınlatılacak her yerin özel bir problem olarak incelenmesi gerekir.

Armatürler, ışık kaynağından çıkan ışığı bir amaca uygun olarak kullanmak için yapılırlar. Genel olarak aydınlatma, amacı bakımından fizyolojik, dekoratif ve dikkati çeken aydınlatma olmak üzere üç tipi vardır. Her aydınlatma amacına göre, farklı armatürler tasarlanabilir. Amacın niteliği, ortam koşulları ve ortamda bulunan nesnelerin özellikleri, tasarımın kapsamını belirleyen faktörlerdendir. Tasarlanan armatürlerin özelliklerini belirlemek için fotometrik verilere bağlı eğriler kullanılır. Işık şiddeti dağılım eğrisi bir armatürün fotometrik özelliklerini verebilen göstergidir. Bu dağılım eğrisi yardımı ile armatüre ihtiyaç duymadan armatür hakkındaki fotometrik bilgiler elde edilebilir. Işık şiddeti dağılım eğrisi

Işık şiddeti dağılımlarını çıkartan goniofotometrelerde fotometrik büyüklüğün açısal değişimine bağlı incelenmesini daha sağlıklı ve hızlı yapabilmek için bu sistemin bilgisayar ile kontrolüne ihtiyaç vardır. Bu çalışmada goniofotometre sistemi adım motorlarının hassas olarak hareket edebilme ve sayısal olarak kontrol edilebilme özelliklerinden faydalanılarak mekanik hareketlerin sağlanması için iki adet melez yapılı adım motoru kullanılarak geliştirilmiştir.

Dünyada ışık şiddet eğrileri çıkartan goniofotometreler hakkında bir çok çalışma yapılmıştır. Bu çalışmalar C.I.E (Commission Internationale d'Eclairage), I.E.S (International Engineering Society) kurumları tarafından standartları hazırlanmıştır. Bu uygulamada ışık kaynağını CIE'nin 70 nolu standartındaki döndüren sisteme sahip goniofotometre modeli temel alınmıştır (1). Goniofotometrenin yapısı tip-3'e göre tasarlanmıştır. Tip-3'ün ışık kaynağına yaptırdığı düzenli hareketler adım motorları ile sağlanmıştır (2). Tutma momentinin daha iyi olması için iki faz uyarımı kullanılmıştır. Motorların kontrolleri paralel port üzerinden yapılmaktadır. Lüksmetre seri porttan RS-232 ile yazılıma veri aktarır. Veri alışverişi ve ışık şiddeti dağılım eğrisi görsel hale getiren ve sanal uygulama alanlarında kullanan yazılım C++ Builder 5.0 bilgisayar programlama dilinde gerçekleştirilmiştir.

Yapay aydınlatma gün ışığının olmadığı ya da yetersiz kaldığı ortamlarda öncelikle iyi görme koşullarının oluşmasını sağlamaktadır. Büyük oranda aydınlatma tekniğine dayalı olarak elde edilebilen iyi görme koşulları, aydınlatma konusunun özelliklerine göre de mimari ve sanatsal açıdan, bir anlamda estetik yönden başarılı görüntüler sunmalıdır (3). Bu nedenle yapılan aydınlatma tasarımlarında konu ile eğitim görmüş uzmanların rol alması kaçınılmazdır. Teknolojik gelişmelerin aydınlatma tasarım ve uygulamalarında kullanması gerekir. Bilgisayarların bilgi işleme kapasitelerinin sürekli artması ile görsellik katarak anlamayı kolaylaştırma, benzetimler sayesinde gerçeği yakalama özellikleri çok geliştirilmiştir. Aydınlatma endüstrisinde kullanılan öğeler matematiksel kavramlar haline getirilerek bilgisayar ortamlarında geliştirilmiştir. Aydınlatma öğelerinin bilgisayar ortamlarına aktarılması ile gerçek hayatı taklit etmek için hazırlanan bilgisayar ortamlarında daha gerçekçi görüntüler oluşturulmuştur. Aydınlatma bilim dalı konuları bir çok bilim dalıyla ortak kullanılmaktadır. Böylece ortak konuları olan bilim dallarıyla bilgisayar ortamında birleştirilerek gerçek hayatta daha verimli olarak kullanılacak endüstri ürünleri ve tasarımları sunulacaktır.

Bu çalışmada, kurulan sistem bilgisayar ortamında oluşturulan çevrenin ve nesnelerin görünebilmesi için yapılan, gerçek armatür verilerinden elde edilen sanal armatür ile gerçekçi görüntü oluşturmaya katkıda bulunmaktadır.

Sanal ortamda hazırlanan bir mekan içerisine goniofotometre sistemi ile gerçek bir armatürün ışık şiddet verileri yazılımla yüklenerek bir armatürün elde edilmiş üç boyuttaki ışık şiddeti verileri doğrultusunda gerçekçi bir aydınlatma tasarım programı yapmak hedeflenmiştir.

Bilgisayar ortamında aydınlatma tasarımlarının yapılabilmesi için mekan, nesneler ve armatürlerin matematiksel olarak tanımlanması gerekir. Armatürlerin fotometrik olarak tanımlanması için temel büyüklüklerden biri olan ışık şiddeti verilerinin uzaysal dağılımının matematiksel olarak tanımlanması gereklidir. Yazılım ışık şiddet eğrilerini en küçük kareler yöntemi ile matematiksel olarak tanımlamaktadır. Bir interpolasyon metodu çeşidi olan en küçük kareler metodu, goniofotometrenin yatay ekseninde aldığı ışık şiddet verilerinden eğri çizdirilmesi ve en iyi yaklaşımı sağlayan matematiksel denklemlerin bulunmasında kullanılmıştır. En küçük kareler metodu ile ışık şiddeti vektör büyüklüklerine göre abak üzerine işaretlenen noktaları istenilen dereceden bir polinom fonksiyonu belirlenmektedir. Uzaya dağılan ışık şiddeti eğrilerin yazılımla birleştirmesiyle armatürlerin ışık dağılım yüzeyi oluşturulmuş, böylece sanal bir mekan sınırları içerisinde herhangi bir doğrultudaki ışık şiddeti değeri ulaştığı piksellerin ışıyım değerlerini ışık şiddeti vektörünün değerine göre değiştirerek aydınlatma benzetimi yapmaktadır.

Armatürlerin matematiksel denklemleri, aydınlatma ve armatür tasarım programlarında bir armatürü tanımlama, özelliklerini kaydetme ve diğer fotometrik büyüklüklere çevirme gibi program içi geçişlerde kullanılmaktadır. Bu özellik geliştirilen goniofotometrenin armatürlerin ışık şiddet eğrilerini çıkartan diğer goniofotometre modelleriyle karşılaştırıldığında özgün bir goniofotometre olmasını sağlamaktadır.

Matematiksel denklemlerle armatür tanımlama yöntemi tanımlanan denklem katsayılarının değiştirilerek önce armatürün ışık dağılım yüzeyi üzerine olan etkilerini izleyebilir. Ayrıca sanal bir ortamda da bu sistemle en uygun armatürü tasarlama veya o ortam için en uygun armatürü seçme işlemlerinde geliştirilen sistem kullanılmaktadır.

Yazılım verileri ile gerçek mekanları birleştirmek için yazılım içinde mekan oluşturma kriterleri belirlemek yerine mekan tasarımı ve animasyon programlarından en çok kullanılan autodesk grubu paket programları çıktılarının kullanılması sağlanmıştır. Autodesk yazılımı 3D studio max paket programı ile özgün olarak tasarlanan sahne “.ase” uzantılı olarak kaydedilirse yazılım içerisinde kullanılabilir hale getirilmiştir.

Aydınlatma tasarımı yapılacak olan mekan, sanal ortama 3D studio max programı ile modellenerek aktarılır. Sistemin goniofotometre kısmı gerçek armatürleri veya kullanıcının tasarladığı armatürleri yazılıma matematik denklemlerini çıkararak kaydeder. Aydınlatma tasarımı yapılacak mekanın modeliyle, armatürün matematiksel modeli ışın etkinlik ve ışın izleme metotlarıyla ayrı ayrı yazılım içinde birleştirilerek bilgisayar ortamında aydınlatma tasarımı yapılır.

2. IŞIK ŞİDDETİ EĞRİLERİYLE İLGİLİ ÇALIŞMALAR

Bu bölümde genel olarak ışın izleme ve ışın etkinlik metotlarının gelişimi, bu metotların aydınlatma tasarım programlarındaki yeri, aydınlatma tasarım programlarının karşılaştırılması ve tanıtılması, ışık şiddeti eğrileri ile ilgili tanımlamalar ele alınacaktır.

2.1. Işın İzleme ve Işın Etkinlik Yöntemlerinin Gelişimi

Sistemin goniofotometre kısmıyla elde edilen bir armatürün ışık şiddeti verilerini 3D studio max paket programı ile tasarlanan sanal mekanlarda gerçekçi bir aydınlatma yapabilmesi için ışık şiddeti verilerini aktarmada ışın izleme ve ışın etkinlik yöntemleri kullanılmıştır. Bilgisayarların kapasite ve bilgi işleme hızlarının sürekli artması bir çok sektörü etkilemektedir. Bu gelişim bilgisayarların getirdiği kolaylıkların başında olan gerçek dünyanın bilgisayar ortamında görselleştirilmesi ve etkileşimli olarak model hesaplama algoritmalarının gelişmesini sağlamıştır. Işığın bilgisayar ortamında benzetimi için kullanılan yöntemlerin başında ışın izleme ve ışın etkinlik gelmektedir.

Işın izleme algoritmaları bilgisayar ortamında gerçekçi görüntü üretimi için 80'li yıllarda ivme kazanmış temel tekniklerden birisidir (4). Gölge, yansıma ve kırılma tanımlanarak kapalı bir alan içindeki aydınlatma modeli ortaya konulmuştur (5). Bu sonuçlar foto gerçekçiliğe ilginin artmasına neden olmuştur. Bu teknikte basit olarak görüntüsü üretilmek istenen manzara ya NURBS (non-uniform rational B-spline) ya da poligonal ağ (poligonal mesh) yüzeylerle modellenmiş nesnelerden oluşturulur. Bu manzara görüntü penceresindeki her piksel'e karşılık gelecek ışınlarla aydınlatılır. Bu ışınlar geometrik optik kuralları dahilinde takip edilirler ve bu ışınların yüzey modelleriyle hangi noktalarda kesiştikleri, hangi ışık kaynakları tarafından oluşturuldukları, hangi alanlarda gölgelerin meydana geldiği gibi bilgiler elde edilir. Bu bilgiler kullanılarak görüntü penceresindeki gerçekçi görüntü oluşturulur.

İlk ışın izleme algoritmalarında, koni izleme stratejisi adı verilen bir yöntemle keskin gölgeler oluşturularak benzetimler yapılmıyordu (6). Koni izleme stratejisi yöntemi tüm nesneleri aynı özellikte kabul etmektedir. Yansıma, kırılma ve şeffaflık gibi tanımlamalara yer vermez.(7). Bu durum yüzeyler arasında ışınların yayılması ve birbirine yansıması hesaplandığında global aydınlatma modelinde hatalı hesaplamalara neden olmaktadır. Lokal yansıma modellerine dayanan daha doğru fiziksel teknikler Blinn (8), Cook (9) ve Torrance (10) tarafından, ışınım alan ısı transferi ve bilgisayar mühendisliği alanlarında geliştirildi.

Işın izleme metodu ışığın bilgisayarda kullanılması ve yansımalarının hesaplanmasında kullanılan ilk metodlardan biridir ve günümüzde 3. nesil aydınlatma metodları olarak da bilinen Global aydınlatma metodlarının temelini oluşturmuştur (11).

Işın izleme yöntemi, varsayılan ışık ışınlarının İki ışık şiddeti eğrisi ile temsil edilebilen armatür bir yol izleyerek etrafı aydınlatması ve görüntü oluşturması yöntemidir. Bu yöntemle son zamanlarda oldukça gelişen tekniklerin ışın içine girmesiyle gölge kontrast ayarı, kırılma, yansıma, gibi karmaşık durumların çözümünde ışın izleme yöntemi rol almıştır. Işın izleme metodunun sınırlılıklarından dolayı ışın etkinlik metodu geliştirildi.

Işın etkinlik metodu 1950'lerde yüzeyler arasındaki radyasyon aktarımını hesaplamak amacıyla geliştirildi (12). 1984 yılında Nishita ve Nakamae tarafından oluşturulan ilk ışın etkinlik çözümleri görüntü sentezinde uygulanmaya başlandı (13). Genel olarak ışın etkinlik metodları dağınık yüzeyler arasında ışığın iç yansımalarını çözmekte kullanılmaktadır. İlk algoritmalar tüm yüzeylerin birbirini görebildiği ortamlarla sınırlı kalmışlardır. Takip eden yıllarda ise, ışın etkinlik algoritmaları gelişti ve ortamın birbirinden bağımsız olarak parçalara ayrılması ile hassaslığı arttı. Işın izleme, bir ortamdaki yüzeylerin yayılan ışığı yutmaları veya aralarında yansımalarının hesabı üzerine kurulu bir yöntemdir. Işığın diğer yüzeylerden yansımasıyla, yüzeyler arasında renk geçişleri ve yumuşak gölgeler gibi önemli aydınlatma etkileri oluşturulan görüntüde izlenir. Bakış doğrultusundan bağımsızdır.

Böylece ışın etkinlik metodları genellikle bakış-bağımsız teknikler olarak adlandırılırlar. Orijinal ışın etkinlik metodu Lambert dağınık yansıma varsayımına dayanırken, sonraki çalışmalar tam yansımayıda kapsayan ışın etkinlik yaklaşımlarını da içine almıştır. 1988'de Cohen, bilgisayar ortamında görüntü oluşturma için daha hızlı çözümlere izin veren "ilerici hassaslaştırma" (progressive refinement) yaklaşımını ortaya attı (14). 1991'de Hanrahan, bütün bir hiyerarşik ışın etkinlik sistemine ait formülü oluşturdu (15). Bütün bu gelişmeler yüzeylerin tanımlanması ve ayrıştırılmasında ciddi bir adım olarak ortaya çıkmıştır.

Her iki yöntemin birbirlerine göre avantaj ve dezavantajları bulunmaktadır. Bir sanal ortama her iki yöntemde uygulandığında bir metotla izlenemeyen özellikler diğer yöntemle görülebilmektedir (16).

2.2. Aydınlatma Tasarımında Kullanılan Programlar

Armatürlerin ışık şiddeti eğrileri temel alınarak yazılmış olan programlardan elde edilen sekiz tanesinin özellikleri incelenmiştir. Aydınlatma tasarımında kullanılan bilgisayar programları arasında bilgisayarlar ile geniş kapsamlı tasarım imkanı sağlayan, Lighting Technologies firması tarafından sürümü yapılan Lumen Micro 7, Zumtobel firmasının ürettiği Cophos, Lumwin ve Dialux programları ile; bu programlarla karşılaştırıldığı zaman daha kısıtlı tasarım olanağı veren Philips firmasının Calculux firmasının aydınlatma programları bulunmaktadır (17).

Mevcut aydınlatma programlarının büyük bir bölümü, aydınlatma tasarımcısını sadece firmanın ürettiği ve piyasaya sunduğu lamba ve armatür tiplerini kullanmaya teşvik etmekte olup programa dışarıdan veri girme imkanı vermemektedir. Bu programlar kısıtlı tasarım imkanları sunmalarına rağmen hemen hepsinde firmanın tasarladığı özgün armatür lamba ve armatürlere ait bilgiler verilmektedir. Bu bilgiler arasında lamba boyutları, lamba ışık akıları, lamba tipleri, lamba güçleri, lamba gerilimleri, renk geri verim özelliği, renk sıcaklığı, armatürlerin ışık şiddeti eğrileri ve bunlar kullanılarak yapılacak hesaplamalar için tablolar yer almaktadırlar.

Dışarıdan kısmen yeni girdi imkanı sunan Zumtobel Luxwin programı ve CAD programları ile birebir uyumlu çalışan ve istenen özellikte ve sayıda lamba veya armatür girdisi olanağına sahip Lumen Micro ve PRC Krochmann firmasının ürettiği programlar türü aydınlatma programlarından tam anlamı ile yararlanabilmek amacıyla, program kütüphanelerine katılacak olan yeni lamba ve armatür özellikleri mekan içindeki istenen her açı ve noktadaki aydınlatma düzeylerinin elde edilmesi için yeterli olmamaktadır. Lamba ve armatür ışık şiddet eğrilerinin tam olarak bilinmesi ve hazırlanan alana bu verilerin girilmesi gerekmektedir. Verilerin farklı açı ve metotlara göre alınması yüzünden genellikle program formatına uygun değerler girilemez.

Bu çalışmada, geliştirilen programla armatürün ışık şiddet veri değerleri her açıya göre tablolarda saklanmaktadır. Bu veriler iki boyuta bir derecesi kullanıcı tarafından belirlenen eğri denklemleri ile tanımlanmaktadır. Geliştirilen programda armatürün verileri girilmesi gerektiğinde denklemi yazılarak eksiksiz olarak tüm veri alt yapısı programa aktarılır. Bu veriler istenirse değiştirilerek o ortama uygun armatür tasarımı için kullanılabilir. Armatür tasarımı denklemlere verilerden eğri oluşturan denklemleri oluşturan yazılım denklem içinde derecesi ve sayısı değiştirilerek aydınlatma tasarımı yapılabilir.

Lightscape R3.2 gerçekçi aydınlatma hesaplarına ve görselleştirme çalışmalarına yeni bir boyut getirmekle birlikte şimdiye kadar yapılmış olan bu tür programların gelişmiş özelliklerini de taşımaktadır. Mimarlar, dekorasyonla uğraşanlar, aydınlatma tasarımcıları ve bilgisayarla gerçekçi resimler elde etmek isteyenler bu programdan geniş ölçüde yararlanmaktadırlar.

İncelenen yaygın olarak kullanılan sekiz programla karşılaştırıldığında ışın etkinlik ve ışın izleme yöntemleri bu program içeriğinde kullanılmaktadır. Program içeriği ile aydınlatma tasarımı yapılmasına imkan verirken içerisinde tanımlanan ışık kaynakları ve gün ışığı seçenekleri sunmaktadır.

3D studio max paket programı tasarımlarda en çok kullanılan programlardan biridir. Özellikle iç mimari tasarımcıları, iç mimaride kullanılan malzemelerin üreticileri bu programla tasarladıkları ortamı bilgisayar ortamında müşterisinin beğenisine sunabilmektedir. Program içeriğinde bulunan tasarım araçlarının gerçekçi görüntüler oluşturması, ışık kaynakları ve gün ışığı seçenekleri, iki ve üç boyutlu sanal gerçeklik alanı vardır (18). Bir nesne üzerine düşen ışık şiddeti sabitlenerek farklı ortamlarda kullanılması veya uzaklığa göre aydınlık düzeyinin ayarlanabilmesi gibi üstün özellikleri vardır. Bu programın geliştirilen sistem içerisinde kullanılabilmesi kurulan sistemin söz konusu üstünlükleri paylaşmasını sağlamıştır.

2.3. Işık Şiddet Eğrileri İle İlgili Tanımlamalar

Birimler sistemin yedi temel birimden biri olan ışık şiddeti, diğer fotometrik büyüklüklere kolayca çevrilebilir. Işık şiddeti bir armatürün fotometrik özelliklerini belirleyen doğrultuya bağlı bir büyüklüktür. Dünyada ışık şiddet eğrileri çıkartan goniofotometreler hakkında bir çok çalışma yapılmıştır. Bu çalışmaların standartları CIE ve IES kurumları tarafından hazırlanmıştır. Bu standartlar incelendiğinde yansıtıcı tiplerinin belirlenmesi (19), fotometrik ölçümler (20), fotometrik başlık seçimi (21), yüksek ve alçak şiddetli ışık kaynakları (22) ve (23), ölçüm cihazlarının ölçüm hızları için kullanılan standartlar dikkate alınmıştır (24).

2.3.1. Işık akısı

Işık akısı Φ sembolü ile gösterilir. Birimi lümen dir. Kısaca lm ile gösterilir. Gündüz görmesine (fotopik görme) ait standart gözün spektral duyarlılık etki eğrisi esas alınır, spektral duyarlılık eğrisine göre değerlendirilen enerji akısına ışık akısı denir (25).

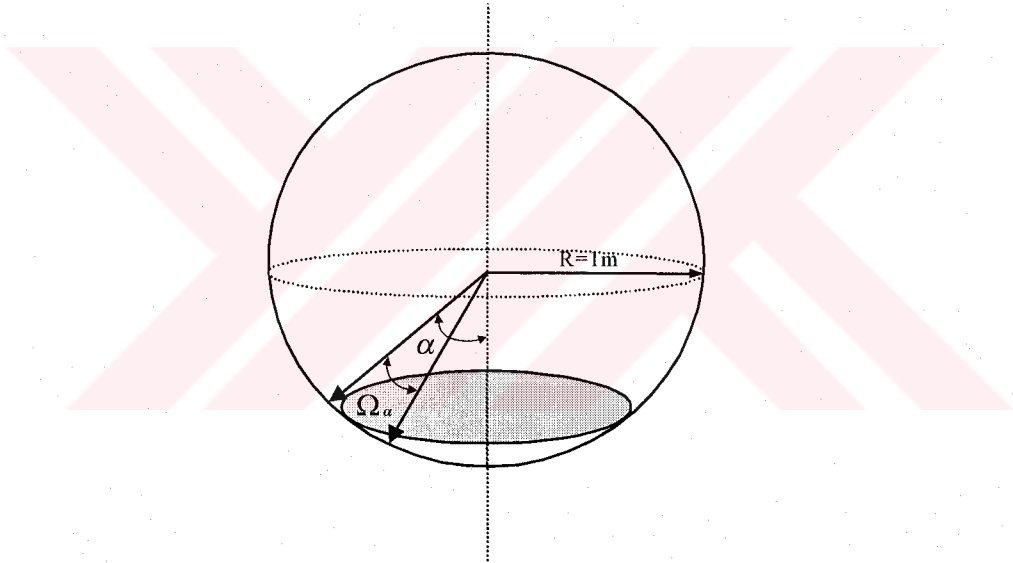
Işık akısı, uzayda yayılırken muhtelif doğrultularında yoğunluğu farklıdır. Bu durum uzay açısı kavramı içinde açıklanabilir. Genel olarak bir kaynağın toplam ışık akısı, uzayın muhtelif kısımlarına yayılan kısmi ışık akılarının toplamı olarak düşünülebilir.

Uzay açı kavramı formülü, düzlemsel tepe açısı α olan bir koninin aştığı Ω_α uzay açısı yarıçapı bir birim olan kürede ele alındığında bu bağıntıdan 0'dan α 'ya kadar olan integrali ile istenilen uzay açısı bağıntısı bulunmuş olur. Şekil 2.1'de düzlemsel açığa ait uzay açının birim küre üzerinde gösterilmesi eş.[2.1]'deki gibidir.

$$d\Omega_\alpha = 2\pi \sin\alpha \, d\alpha$$

$$\Omega_\alpha = 2\pi \int_0^\alpha \sin\alpha \, d\alpha = 2\pi [-\cos\alpha]_0^\alpha \quad [2.1]$$

$$\Omega_\alpha = 2\pi (1 - \cos\alpha)$$



Şekil 2.1. Düzlemsel açığa ait uzay açının birim küre üzerinde gösterilmesi

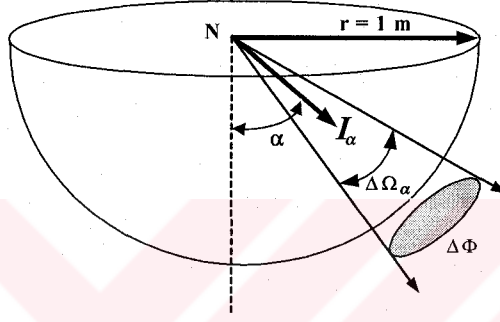
2.3.2. Işık şiddeti

Bir ışık şiddeti eğrisi ile temsil edilebilen armatür bir ışık kaynağının herhangi bir α doğrultusundaki ışık şiddeti bu doğrultuyu içine alan bir $\Delta\Omega$ uzay açısından çıkan $\Delta\Phi$ ışık akısının $\Delta\Omega$ uzay açısına bölümü ile ilgilidir. Eğer ortamın ışığı yutma özelliği yoksa,

$$I = \frac{\Delta\Phi}{\Delta\Omega} \quad [2.2]$$

oranı ortalama ışık şiddeti olarak tanımlanır. $\Delta\Omega$ sıfıra yaklaşırken bu oranın limiti de I_α ışık şiddetini tanımlar.

$$I_\alpha = \lim_{\Delta\Omega \rightarrow 0} \frac{\Delta\Phi}{\Delta\Omega} = \frac{d\Phi}{d\Omega} \quad [2.3]$$



Şekil 2.2. Bir ışık şiddeti eğrisi ile temsil edilebilen armatür bir ışık kaynağının α doğrultusundaki I_α ışık şiddetinin tanımlanması

Bu denklemde ışık akısı lümen ve uzay açısı steradyan cinsinden yerlerine konursa, ışık şiddeti kandela cinsinden bulunur. Şekil 2.2’de Bir ışık şiddeti eğrisi ile temsil edilebilen armatür bir ışık kaynağının α doğrultusundaki ışık şiddetinin tanımlanması gösterilmiştir. Buna göre 1 steradyanlık uzay açıdan çıkan ışık akısı 1 lümen ise ışık şiddeti 1 candela olur;

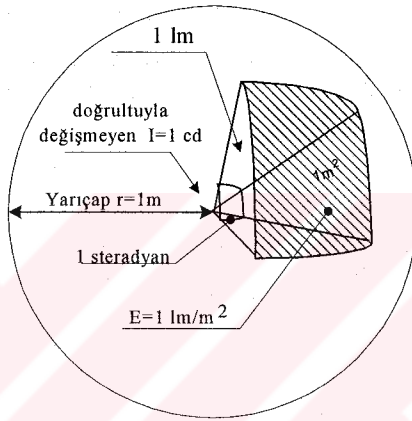
$$I \text{ (cd)} = \frac{\Phi \text{ (lm)}}{\Omega \text{ (sr)}} \quad [2.4]$$

Işık şiddeti I harfi ile gösterilir. Birimi kandeladır. Doğrultuya bağlı bir büyüklüktür.

2.4. Işık Dağılım Küresi

Tüm doğrultularda ışık şiddeti aynı olan bir Bir ışık şiddeti eğrisi ile temsil edilebilen armatür ışık kaynağının, yarı çapı 1 metre olan bir kürenin merkezinde bulunduğu varsayılırsa, bu kürenin iç yüzeyinde düzgün yayılmış bir aydınlık düzeyinin

oluşacağı, ışık ölçümsel büyüklük ve birimler ile ilgili temel kavramları tanımlar. Şekil 2.3'de ışık dağılım küresi gösterilmiştir. Aynı temel tanımlara göre, bir ışık kaynağının doğrultuya göre değişmeyen, yani izotropik özellik gösteren ışık şiddeti 1 kandela (1 cd) ise, bu kaynak her bir steradyan açı içine 1 lm ışık akısı yollar, bu akı küre merkezinden geldiği ve İki ışık şiddeti eğrisi ile temsil edilebilen armatür yayıldığı için bu açının dışına çıkamaz ve bu açının gördüğü küre yüzeyi 1 m^2 olduğundan, küre yüzeyindeki aydınlık düzeyi de $1 \text{ lm/m}^2 = 1 \text{ lüks}$ olur.



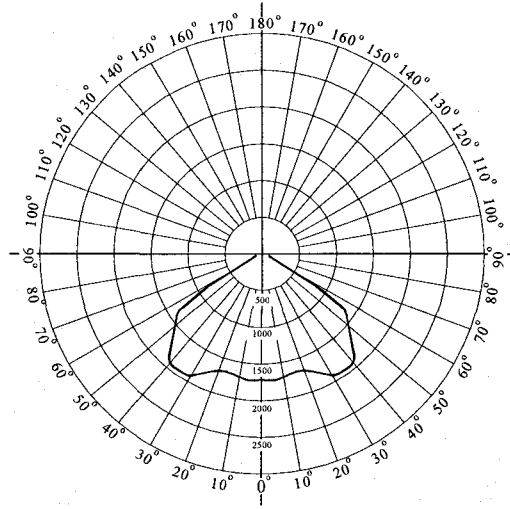
Şekil 2.3. Işık dağılım küresi

2.5. Işık Şiddeti Eğrisi

Bir ışık kaynağından çıkan ışık şiddet vektörlerinin uç noktalarının geometrik yeri ışık dağılım yüzeyini oluşturur. Bu ışık kaynağının merkezinden geçen ve ışık dağılım yüzeyini kesen düzlem ile ışık dağılım yüzeyinin ara kesiti ışık şiddeti eğrisidir. Işık şiddeti eğrisi, bir ışık kaynağının hangi doğrultuya ne kadar ışık akısı yayımladığını gösterir. Bu eğri, açığa göre ışık şiddetlerini cd/klm olarak verir.

2.5.1. Bir ışık şiddeti eğrisi ile temsil edilebilen armatür

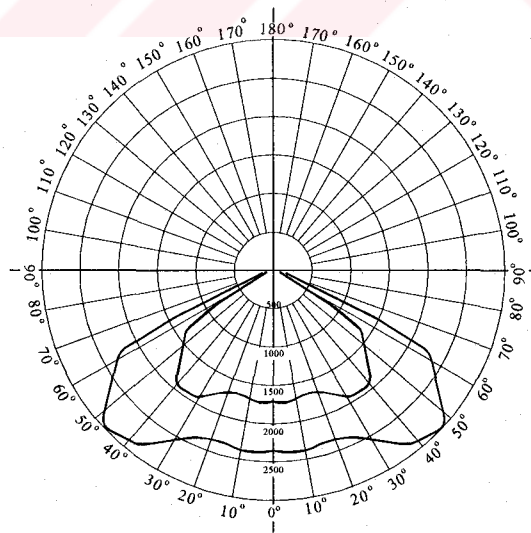
Armatürün kendi etrafında döndürülmesi ile oluşacak olan ışık şiddet eğrileri birbirinin aynısı olduğundan bu tür armatürlerin tek eğrisi vardır Şekil 2.4'de Bir ışık şiddeti eğrisi ile temsil edilebilen armatürün ışık şiddet eğrisi gösterilmiştir.



Şekil 2.4. Bir ışık şiddeti eğrisi ile temsil edilebilen armatürün ışık şiddet eğrisi

2.5.2. İki ışık şiddeti eğrisi ile temsil edilebilen armatür

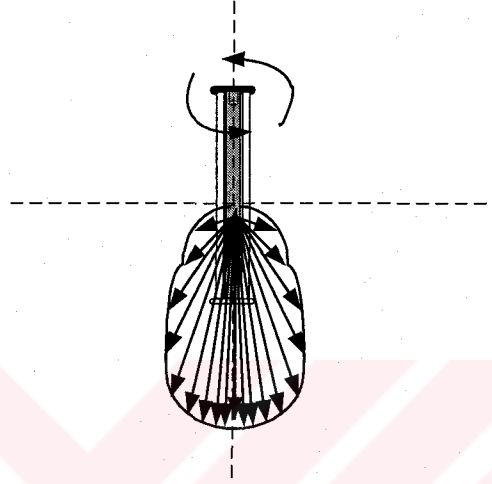
Armatürün fotometrik merkezinden geçen birbirine dik iki düzleme göre farklı ışık dağılımına sahip olan armatürdür. Şekil 2.5’de İki ışık şiddeti eğrisi ile temsil edilebilen armatürün ışık şiddet eğrileri gösterilmiştir.



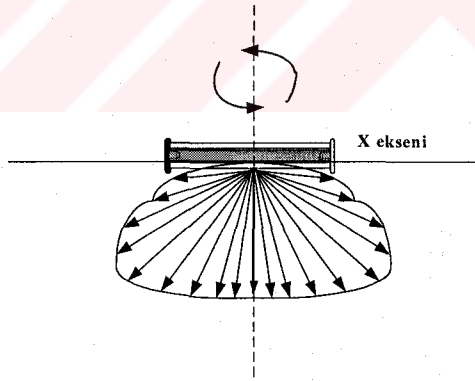
Şekil 2.5. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün ışık şiddet eğrileri

İki ışık şiddeti eğrisi ile temsil edilebilen armatürlerin ışık yayılım doğrultusunda, uzun kenarının yatay eksende yaptığı hareket enine ve kısa kenarının yatay eksene paralel

olarak yaptığı hareket boyuna harekettir. Işık kaynağının boyuna ve enine kesitlerinden geçen düzlemler için iki ayrı ışık şiddet eğrisi çıkarılır. Şekil 2.6'da İki ışık şiddeti eğrisi ile temsil edilebilen armatürün boyuna kesiti gösterilmiştir. Şekil 2.7'de İki ışık şiddeti eğrisi ile temsil edilebilen armatürün ışık şiddet değerleri gösterilmiştir.



Şekil 2.6. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün enine kesiti



Şekil 2.7. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün boyuna kesiti

2.5.3. Üç ve daha fazla ışık şiddeti eğrisi ile temsil edilebilen armatür

Armatürün fotometrik merkezinden geçen birbirine dik düzlemlere göre farklı ışık dağılımına sahip olan armatürdür. Bu armatürün köşegen sayısına göre eğri sayısı belirlenmelidir.

2.6. Işık Şiddeti Dağılım Yüzeyi

Bir kürenin merkezine yerleştirilen ışık kaynağı, tüm doğrultularda ışık şiddeti dağılımına sahiptir. Bu ışık kaynağının küre üzerindeki ışık şiddeti aynı olan doğrultularını gösteren noktaları birleştirilince ortaya çıkan üç boyutlu ışık şiddeti dağılım küresine ışık şiddeti dağılım yüzeyi denir (25).

2.7. Fotometrik Yasalar

Fotometrik yasaların açıklanmasıyla gelecek bölümlerdeki hesaplamalar daha iyi anlaşılacaktır.

2.7.1. Kosinüs yasası

Paralel ışınlı bir ışık demeti içinde bulunan bir s yüzeyinin aydınlık düzeyi yüzeyin normali ile ışık demeti arasındaki açı α olduğuna göre,

$$E = E_0 \cos \alpha \quad [2.5]$$

denklemini ile değişir ve bu bağıntıya kosinüs yasası denir.

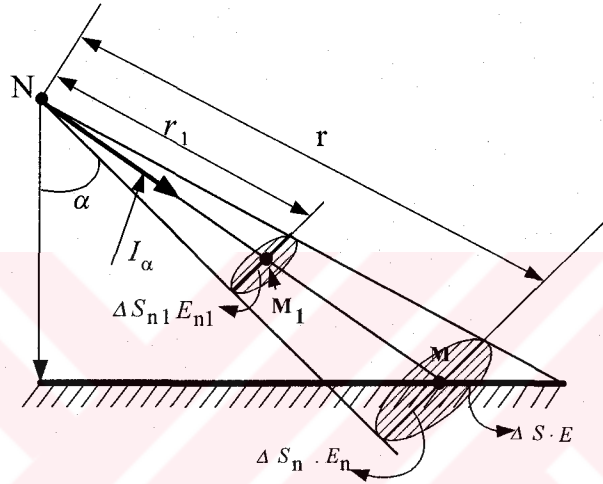
2.7.2. Lambert yasası

Işık yayan bir yüzeyin parıltısı, her doğrultuda sabit ise bu yüzeye lambert yasasına göre ışık yayan veya ideal dağıtıcı yüzey denir.

$$L_\alpha = L = \text{Sabit} \quad [2.6]$$

2.7.3. Uzaklıklara karesi ile ters orantı yasası

Bir ışık kaynağı herhangi bir α doğrultusunda ışık şiddeti meydana getiriyorsa, bu doğrultuya dik düzlemdeki aydınlık düzeyleri, düzlemlerin kaynağa olan uzaklıklarının karesiyle ters orantılıdır. Şekil 2.8'deki fotometrik uzaklık kanunu formülünün çıkartıldığı şekil gösterilmiştir.



Şekil 2.8. Uzaklıklara karesi ile ters orantı yasası prensip şeması

$$\frac{E_n}{E_{n1}} = \frac{r_1^2}{r^2} \quad [2.7]$$

Burada E_n , ΔS_n düzleminin ortalama aydınlık düzeyini, E_{n1} ise ΔS_{n1} düzleminde ortalama aydınlık düzeyini gösterir. $\Delta\Omega$ uzay açılı koniden çıkan ışık akısı ;

$$\Delta\Phi = I_\alpha \Delta\Omega \quad [2.8]$$

Bu ışık akısı r_1 uzaklığında bulunan bir düzlemin ΔS_{n1} alanını aydınlattığı halde, r uzaklığında bulunan bir düzlemin $\Delta S_n = \frac{r^2}{r_1^2} \Delta S_{n1}$ alanını aydınlatmaktadır. Buradan

ΔS_n ve ΔS_{n1} düzlemlerinin aydınlık düzeyleri;

$$\Delta E_{n1} = \frac{\Delta \Phi}{\Delta S_{n1}} \text{ ve } E_n = \frac{\Delta \Phi}{\Delta S_n} \quad [2.9]$$

olarak yazılırsa;

$$\frac{E_n}{E_{n1}} = \frac{r_1^2}{r^2} \text{ veya } E_n = E_{n1} \cdot \frac{r_1^2}{r^2} \quad [2.10]$$

bağıntısı bulunur. Eğer sonucu denklemde E_{n1} yerine $E_{n1} = \frac{I_a \Delta \Omega}{\Delta S_{n1}}$ konulursa ve

$\Delta \Omega = \frac{\Delta S_{n1}}{r_1^2}$ olduğu düşünülürse,

$$E_n = \frac{I_a}{r^2} \quad [2.11]$$

denklemini elde edilir. Kosinüs yasasından faydalanılarak herhangi bir düzlemin aydınlık düzeyi;

$$E_n = \frac{I_a}{r^2} \cdot \cos \alpha \quad [2.12]$$

denklemini ile hesaplanır.

2.7.3.1. Uzay açı izdüşümü yasası

Lambert yasasına uygun ışık yayan bir S yüzeyinin çalışma düzleminin bir P noktasında meydana getirdiği aydınlık düzeyi hesaplanmaktadır. Şekil 2.9'da uzay açı yasası prensip şekli gösterilmektedir.

3. IŞIN İZLEME VE IŞIN ETKİNLİK METODLARININ AYDINLATMA BENZETİMİNE UYGULANMASI

Bilgisayarların günlük hayata girmesiyle bir çok alanda hayatımız kolaylaşmıştır. Hayatımızı kolaylaştırdığı alanlardan birisi de bilgisayarda benzetim olanaklarıdır. Bir uygulamayı daha önceden bilgisayarda hesaplama, görsellik katarak anlamayı kolaylaştırma, benzetim sayesinde gerçeği yakalama çabası sayılabilir. Bu özellikler eğitim, görsel oyun, reklamcılık, mimarlık, film endüstrisi gibi alanların gelişmesine yardımcı olmaktadır.

Bilgisayarların kapasite ve bilgi işleme hızlarının sürekli artması bir çok sektörü etkilemektedir. Bu gelişim bilgisayarların getirdiği kolaylıkların başında olan gerçek dünyanın bilgisayar ortamında görselleştirilmesi ve etkileşimli olarak modelleme algoritmalarının gelişmesini sağlamıştır. Gerçek dünyanın bilgisayar ortamında görselleştirilmesinde ışığın ve görüntünün bilgisayar yardımı ile hesabı geçmişten bu yana en zor ve gerekli algoritmalar olarak öne çıkmaktadır. İşte bu algoritmaların başında ışın izleme ve ışın etkinlik gelmektedir. Bu metodlar global aydınlatma metodlarının temelini oluşturmaktadır.

Işın izleme ve ışın etkinlik metodları son zamanlardaki görüntü sentezlerinin gelişmesine bağlı olarak ortaya çıkmıştır. CRT'ler 1940'larda bilgisayar ekranlarında kullanılmaya başlandı. Bu aygıtlar bilgisayarlarda CRT üzerinde desteklenen koordinatlardan dolayı vektörel çizgiler ve noktaları çizebilme yeteneğine sahiptir. Ivan Sutherland'ın SKETCHPAD programı, etkileşimli iki boyutta çizim uygulaması etkileşimli bilgisayar grafiklerinde önemli bir ilerleme sağladı(26). 1950 'lerdeki serbest biçimli (free-form) yüzeylerin tanımlanmasındaki gelişmeler vektörel grafiklerin gelişmesini sağladı. Vektörel grafiklerin gelişmesiyle bilgisayarların endüstriyel alanlarda kullanılmasına başlandı. 1960'lardan sonra, tarama grafikleri bu uygulamaların sonucunda ortaya atılmıştır. Piksellerdeki bir dizi için, bilgisayarda tanımlı renklerden oluşan tarama grafikleri, vektörler yerine kullanılarak, yüzeyler için çok daha gerçekçi bir görünüm sağladı. Bouknight tonlar arasındaki farkların

tanımlanmasını ve grafiklerde kullanılmasını sağladı (27). Gouraud yüzeyleri eğrilerle tanımladı (28). Phong yüzey şekillerini görsel bir şekilde karakterize etmek için gölge modellerinin kullanımını çalışmalarıyla geliştirmişti (11). Fakat bu modeller; ışık yansımalarında kullanılan fiziksel modeller değillerdi. Bu modeller göz, yüzey ve ışığın bağıl pozisyonlarını kullanan gölgeleme hesaplarıydı. Diğer yüzeylerdeki ışık yansımaları gibi aydınlatma biçimleri hesaba katılmıyordu. Aynı şekilde bir yüzeyden diğerlerine gölge düşürme gibi global olgular da hesaba katılmıyordu.

İlk modellerde nesnelerin tam olarak gerçekte görüldüğü nitelikte ve gölgelenmiş biçimde gösterilmesi hedeflenmemiştir. Gerçek nesnelere mümkün olan en yakın gerçeklik derecesinde bir görüntü sağlanılmasına çalışılmıştır. İlk aydınlatma modellerine direkt ve dağınık ışık aydınlatma etkisi, uzaysal yayılım, yansıma, yüzey dokusu etkileri ve yüzey geçirgenlikleri gibi bir çok problem saptanmıştır. Bu problemleri çözmeye yönelik olarak yapılan çalışmalar kabartma kaplamaları ve doku kaplamaları gibi tekniklerin gelişmesine neden oldular. Bunlar daha karmaşık yüzey özelliklerinin daha gerçekçi olarak bilgisayara aktarılmasıyla sonuçlandı. Gerçekçi görüntü oluşturmada en kapsamlı algoritmalar, ışın etkinlik ve ışın izleme metodlarıdır.

3.1. Işın İzleme Metodu

Bilgisayar ortamında oluşturulacak olan mekan, NURBS yada çokgensel ağ yüzeylerle modellenmiş nesnelerden oluşturulur. Görüntüdeki her piksel için bir ya da daha çok ışın kameradan sahneye yöneltilir ve sahnedeki bir nesne ile kesişip kesişmediğine bakılır. Kameradan çıkan bu ışının bir nesneyi kestiği her noktadaki yüzey rengi hesaplanır. Bunun için ışınlar sahnedeki her ışık kaynağına da yönlendirilir ve böylece kaynaklardan gelen ışık miktarı da hesaplanır. Benzer şekilde her yüzey noktasının gölgede kalıp kalmadığı hesaplanır. Yüzeyin şeffaf olup olmamasına göre de ne kadar ışığı geçirdiği ve ne kadar ışık kırılmasına yol açtığı da hesaplanarak yüzeyin son rengi belirlenir.

Bu yöntem, gelişen bilgisayar endüstrisi ve geliştirilen tekniklerin kullanılmasıyla gölge zıtlık ayarı, kırılma, yansıma, yumuşak gölge gibi karmaşık alanların çözümünde rol almıştır. Gerçekçi bir görüntü elde edilebilmesi için nesnelerden sonsuz sayıda yansıma yapılması gereklidir. Bu hesaplamalar dizisi bilgisayarlarda çok zaman almaktadır. Hesaplama zamanını azaltmak için bazı sadeleştirme metodlarının kullanılması gerekmektedir.

3.1.1. Işın izlemede sadeleştirme teknikleri

Gerçekte bir ışın sonsuz defa enerjisini kaybedene kadar etrafta yansır. Sonsuz defa yansımayı bilgisayar ile hesaplayabilmek mümkün değildir. Ancak bir ışın iki yansıtıcı yüzey arasında çok sayıda irdeleme yapacak şekilde yansıtılır. Bilgisayarda hesaplama yapılırken yansıma sayısı belirli bir sınırdan tutulmalıdır. Benzetimi yapılacak nesnelerden oluşacak olan sahnenin sınırları da belirlenmelidir.

Görünmeyen yüzeyler hesaplama dahil edilmezler. Örneğin bir göz sadece görüş alanındaki çevreyi görür. Bilgisayarda da aynı yöntem kullanılarak görünmeyen yüzeyler ihmal edilir. Bu hesaplamalar yapılırken dikkat edilmesi gereken bir nokta da şudur. Eğer benzetimi yapılacak alanda, yansıtıcı özelliği olan bir nesne varsa bu kameranın görmediği bir noktada olan nesneleri de bir açıdan kameraya yansıtabilir veya görüş alanında olmayan bir nesnenin gölgesi görülebilir.

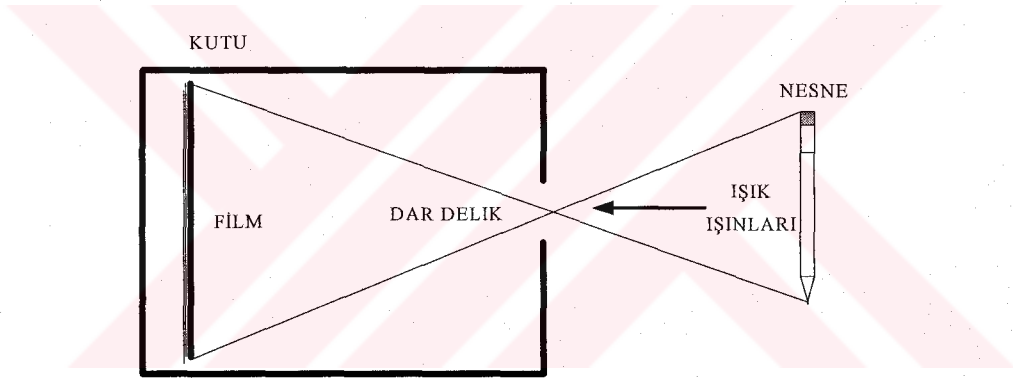
Sadece kameranın görebildiği görüntüye izi düşen ışınlar hesaba alınır. Diğer açıları izleyen ışınlar hesaplamada dikkate alınmayarak sadeleştirme yapılır. Burada açıklanan sadeleştirme algoritmaları ile hesap, çok daha hızlı ve mümkün hale getirilebilmektedir.

3.1.2. Işın izleme ile ilgili tanımlamalar

Işın izleme yöntemi anlatılmadan ışın izlemede kullanılan bazı tanımlamalar aşağıda verilmiştir

Bilgisayar ortamında oluşturulan görüntülere grafik denir. Grafiklerde bakış açısına göz veya kamera denir. Işın izleme metodunda oluşturulacak olan grafikler göze göre hesaplanır. Işın izlemede grafik oluşturma, iğne deliği kamerasıyla oluşturulan grafiğe benzerdir.

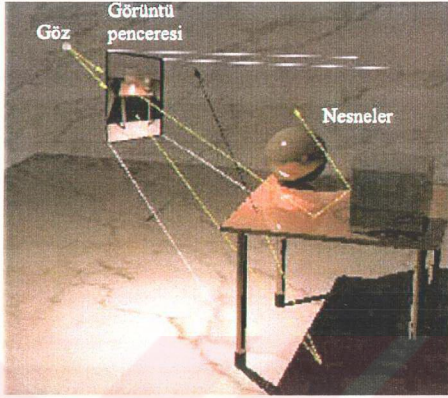
İğne deliği kamerasının çalışması şu şekilde açıklanır. Kapalı bir kutunun içerisine fotoğraf filmi yerleştirilir. Bir toplu iğne ile kutunun ön kısmına bir pencere açılır. Pencere bantla kapatılır. Fotoğraf çekebilmek için kutu sabit tutulup, bant kaldırılır. Işıklar içeri girerek filme çarparlar. Dışarıdaki bütün ışıkların filme çarpmaması için mümkün olduğu kadar küçük bir pencere açılır. Pencere kısa bir süre açık tutulur. Şekil 3.1’ de iğne deliği kamerası gösterilmiştir.



Şekil 3.1. İğne deliği kamerası

Grafiklerde istenilenlerin çizildiği alana görüntü penceresi veya ekran denir. Şekil 3.2’de göz, görüntü penceresi ve nesneler gösterilmiştir. Bilgisayarda işletim sistemi içindeki, denetim masası görüntü ayarlarından ekran alanını oluşturacak piksel sayısı belirlenebilir.

Işın izleme metodunda temel amaç, görüntü penceresindeki piksellerin rengini, ışık kaynağı ve nesnelere göre hesaplamaktır. Bir bütünün parçaları olan bu piksellerin belirlenen renkleri grafiği oluşturur.

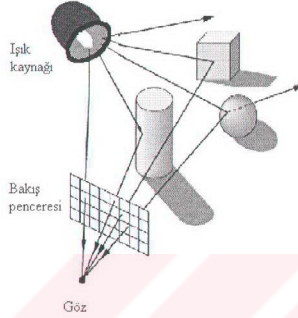


Şekil 3.2. Göz, görüntü penceresi ve nesneler

Görüntü oluşturulurken, görüntü penceresinde resmi oluşturulan tüm nesnelerin matematiksel bir tanımı ve koordinat sistemi üzerinde sabit bir yeri olmalıdır. Işın izlenmede görüntü penceresi, matematiksel olarak tanımlanabilen küp, prizma, küre, düzlem gibi nesnelerden ve bu nesnelerin birleşerek meydana getirdiği karmaşık yapıları nesnelerden oluşturulur. Görüntü penceresi, nesneler ve nesnelerin gerçekçi bir şekilde görüntülenebilmesi için doğrultusu bilinen ışıklardan oluşturulmalıdır. Işık kaynaklarının yerleri ve ışık şiddeti değerleri, çevrelerindeki nesnelerin gerçekçi bir şekilde benzetimi için önemli özelliklerdir. Geliştirilen ışın izleme metodunun diğer ışın izleme uygulamalarından farkı, sistemin goniofotometre kısmıyla elde edilen ölçümlere bağlı armatürlerin ışık şiddeti verileriyle görüntü elde etmesidir.

Kamera ile gerçek bir alanı izlerken şu tespitler yapılabilir; ışık kaynağından çıkan ışınlar sonsuz sayıda yansıma yaparak çevreye dağılırlar. Bu çevreye dağılan ışınların bir kısmı kameraya doğrudan ulaşırken geriye kalan ışınlar dolaylı olarak kameraya ulaşır veya başka bir yöne yansyarak görüntüden çıkarlar. Benzetimi yapılacak alanda ise bu tespitleri kullanarak görüntü penceresi içerisinde yansıma yapacak ışınlar hesaplamaya katılır. Görüntü penceresi içerisinde yansıma yapacak ışınlar şu şekilde belirlenir. Işık kaynağından çıkan ışınları izlemek yerine kameradan

başlayarak ışınlar, ışık kaynağına kadar izlenir. Bu ışın izleme metodunun temel yaklaşımıdır. Şekil 3.3’de ışık kaynağından çıkan ışınların göze doğru yansımaları gösterilmiştir.



Şekil 3.3. Işık kaynağından çıkan ışınların göze doğru yansımaları

Işınlar, benzetim alanı içerisindeki matematiksel olarak tanımlanmış nesneleri ışık şiddeti değerlerine bağlı olarak etkilerler. Bu etkileşim sonunda bakış penceresinde dolaşan ışının renklerine göre nesnelerin renkleri hesaplanır. Işının göze çarpmasından önce çok sayıda yansıma olduğu için ışınları izleyen yansımalara bir sınır koymak gerekir.

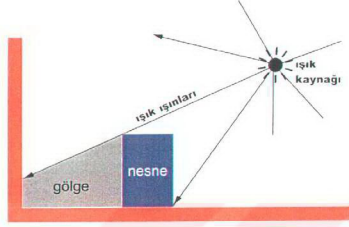
3.1.3. Işın izleme metoduyla görüntü oluşturma

Işın izleme metodu ile görüntü oluştururken gerekli tanımlamalar ve görüntü oluşumunun aşamaları aşağıda verilmiştir.

3.1.3.1. Gölgeleme

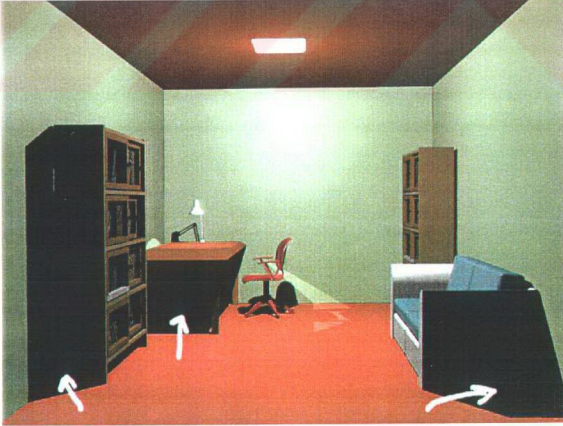
Işık kaynağından çıkan ışınlar düz bir yol izleyerek nesnelere çarparlar. Bu çarpış sırasında nesnenin yansıtma ve geçirgenlik özelliklerine bakılarak bu ışınlar nesneler tarafından kesilirler.

Işığın tutan nesnelerin arkasında ışınların ulaşamadığı karanlık bir bölge oluşur. Şekil 3.4'de gölgelerin oluşması gösterilmiştir. Nesnenin arkasında kalarak doğrudan ışık kaynağının ışınlarını alamayan bölgeler belirlenmiştir.

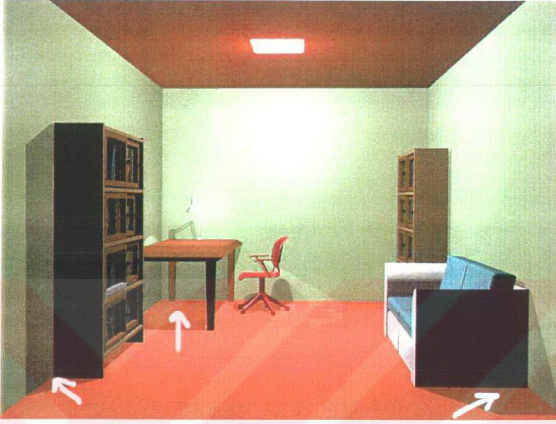


Şekil 3.4. Gölgelerin oluşması

Işın izleme metodunda, ışınların yansıması ve nesnelerin konumuna göre gölgelerin karanlık seviyeleri ve kenar yumuşaklığı ayarlanabilmektedir. Şekil 3.5'deki sahnede bulunan tam gölge oluşturma, şekil 3.6'da ise (%50) yarım gölge oluşturan ışık şiddetine göre gölge yumuşaklıkları gösterilmiştir.



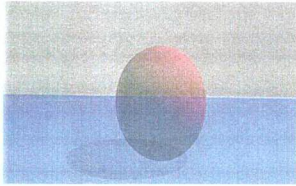
Şekil 3.5. Işın izlemede tam gölge oluşturma



Şekil 3.6. Işın izlemede yarım gölge oluşturma

3.1.3.2. Şeffaflık

Şeffaflık nesnenin içinden ışığı geçirebilme, dolayısıyla ışığın geçebildiği bu bölgenin arkasındaki alanın belli oranlarda görülebilmesidir. Gerçekte şeffaflık seviyesi olan hiçbir nesne, ışığı kırmadan geçirmez. Işık şiddeti artırılırsa şeffaf nesnelerin gerisindeki nesneler daha rahat görülür, bunun terside doğrudur. Şekil 3.7’ de yarı şeffaf malzeme arkasından gözlenen bir nesne ışın izleme metodu ile kullanılarak oluşturulmuştur.



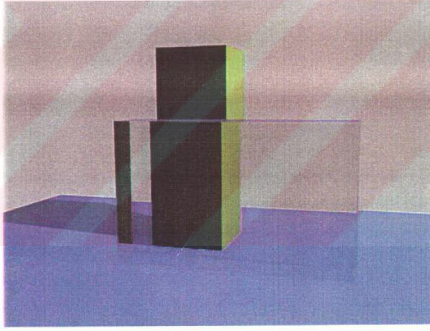
Şekil 3.7. Yarı şeffaf malzeme arkasından gözlenen bir nesne

Eğer yüzeyin diğer tarafında farklı maddeler farklı ışık kırma indislerine sahipse; örnek olarak bir taraf hava diğer taraf su ise ışın kırılmış olacaktır. Aynı madde yüzeyin her iki tarafında da varsa kırılan ışın orijinal ışın gibi aynı yönde ilerler ve kırılmaz.

3.1.3.3. Kırılma

Nesnelerin, üzerine gelen ışınları yoğunlukları oranında kırarak farklı açılarla gönderme olayına kırılma denir.

Şeffaf nesnelerin yoğunluklarına göre farklı indisleri vardır. Şekil 3.8’de kalın bir camın yoğunluğuna göre ışık indisi özelliği ışın izleme ile tasarlanarak gösterilmiştir.

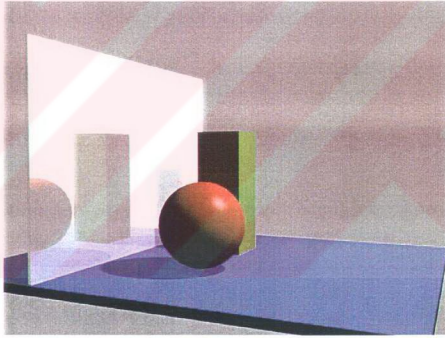


Şekil 3.8. Kalın bir camın yoğunluğuna göre ışık kırma indisi özelliği

3.1.3.4. Yansıma

Bir ışın bir nesneye çarptırıldığında, yazılım ışın izleme yöntemi ile o noktanın rengini belirlemelidir. Yüzeyin rengi nesneden yansıyan ışın ile başka bir nesnenin rengini de belirlemelidir. Bunu bir ışın izleme algoritması ile modelleyebilmek için ışının yansıyacağı açıyı hesaplamak gerekir. Bu nesnelerin yapıldığı malzemeye göre

bilgisayar programında tanımlanır. Bilgisayarda ışık kaynağının ışık şiddetine bağlı olarak ışınların yapmış olduğu yansımalar nesnelerin renklerini etkilemektedirler. Bunun için hesaplanan yönde kesişim noktasından yansıyan başka bir ışın oluşturulmalıdır. Öncelikle nesnenin görünüş renginin ne olması gerektiği bilindiğinden ilk renk önceden verilir. Yansımada nesnelerden çoklu yansımalar; kırılmada ise bir ışının saydam nesnelerden parça parça veya tamamen geçtiğinde ne olacağı hesaplanır. Eğer nesne yansıtabilirse yansımış ışın kesişme noktasından yansıma yönüne doğru izlenir. Yansımış ışın orijinal ışının aynasal yansımasıdır yani simetrisidir. Eğer nesne saydam ise yansıyan ışın yüzeyin içinde izlenir. Şekil 3.9'da ışın izleme metodu kullanılarak küre ve dikdörtgenler prizmasının aynadaki yansımaları görülmektedir.



Şekil 3.9. Küre ve dikdörtgenler prizmasının aynadaki yansımaları

3.2. Işın Etkinlik Metodu

Işın etkinlik, sayısal çözüm ve yaklaşımlarıyla bir enerji denge eşitliğidir. Işın izlemede sadece bakış penceresindeki yansımalar dikkate alınırken, ışın etkinliğin üstünlüğü çevredeki nesnelerin üzerindeki yansımaların tamamını değerlendirmeye olarak daha gerçekçi bir çözüm oluşturmaktadır. Pek bilinmeyen özelliklerinden biri de görüntü konumunun bağımsız olmasıdır ve bundan dolayı ışın etkinlik metodu

bakış doğrultusundan bağımsız (view-independent) bir grafik sunar. Işın etkinlik yöntemi ile elde edilen görüntü kalitesinin sınırları, kullanılan bilgisayarın ekran çözünürlüğüne ve hesaplama hızına bağlıdır. Ekran çözünürlüğü ve hesaplama hızı yüksek bir bilgisayar ile grafiğinin içerisindeki bakış açısı, nesneler, odak, ışık kaynaklarının yeri ve büyüklükleri değiştirilebilir, aynı zamanlı benzetim ile grafik içi gezinti yapılabilir. Son yıllarda bilgisayar grafiklerinde sıkça kullanılmaktadır. Oyun sektöründe hızlandırılmış ve sadeleştirilmiş algoritmaların üretilmesi ve kullanması bu alanda bir devrim olarak kabul edilmektedir. Bu metod yeni nesil global aydınlatma benzetimi metodudur.

Işın etkinlik metodu ışın izleme metodundan daha gerçekçi benzetimler yapılabilir. Aynı zamanda gelişmişlik düzeyi ile doğru orantılı olarak bilgisayarda hesaplama daha uzun sürer. Işın etkinlik metodu gerçek dünyadaki ışınların yansıma prensibine çok benzerdir ve bu prensibi taklit eder.

Işın etkinlik metodunda, her ışık kaynağından çıkan ışınlar etrafa gelişi güzel dağılır. Bu ışınlar çarptıkları yüzeylerde enerjilerinin bir kısmını bırakırlar. Bu yüzeyde artan enerji kendinden aydınlatma oluşturur ve bu yüzey aydınlanmış olarak görülür. Bu yüzeyden enerjisi azalarak yansıyan ışınlar doğrultularını değiştirerek başka yüzeylere çarparlar ve bu gerçek dünyada sonsuza dek sürer. Nesnenin yapısına göre ışınlar; yansıma, kırılma, şeffaf yüzeylerden geçme gibi özelliklere sahip olurlar.

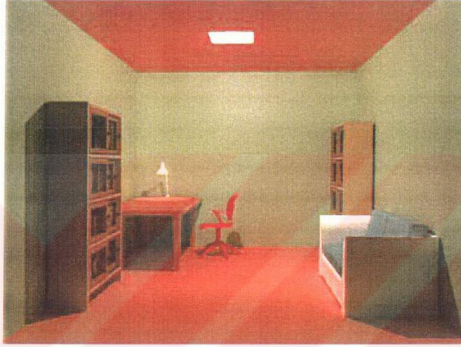
3.2.1. Işın etkinlik ile görüntü oluşturma

Bu kısımda ışın izleme metodu ile görüntü oluştururken gerekli tanımlamalar ve görüntü oluşumunun aşamaları verilmiştir.

3.2.1.1. Gölgeleme

Işın etkinlik metodunda gölgeler hiçbir zaman tam gölge (%100) değildir. Çünkü her zaman ışınlar bu gölge bölgelerine diğer yüzeylerden yansıyarak gelirler ve enerji bırakırlar. Sadece gölge kısımlara daha az yoğun olarak çarparlar.

Işın etkinlik metodunda “karma renk” ve “yayılnma” adı verilen algoritmalar bulunur. Bu algoritmalarda rengin ve ışığın dağılımı söz konusudur. Şekil 3.10’da ışın etkinlik metodu kullanılarak aydınlatması hesaplanmış bir oda gösterilmiştir.



Şekil 3.10. Işın etkinlik metodu kullanılarak aydınlatması hesaplanmış bir oda

Nesnenin rengi ve üzerine düşen ışınların ışık şiddeti değeri nesnenin görünen rengini belirler. İlk ışık şiddeti değerini gerçek verilerden alması hazırlanan ışın etkinlik algoritmasının daha gerçekçi hale gelmesi sağlar. Bu metod daha gerçekçi yumuşak geçişi olan gölge bölgelerine imkan verir.

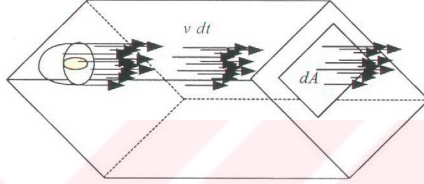
3.2.2. Işın etkinlikte sadeleştirme teknikleri

Gerçek dünyada ışınlar sonsuz sayıda yansımaya uğrarlar, fakat bilgisayar grafiklerinde bu yansıma değerini sınırlamak gerekir. Görünmeyen bölgelere ışın gönderilmez. Gönderilen ışınların sayısı gerçek dünyada sonsuz sayıda olmasına rağmen bilgisayar hesaplamalarında bu sınırlandırılmıştır.

Yüzeyler arasındaki form faktörü değeri küçük değerlere ulaştığında program yansıma sayısını durdurur.

3.2.3. Işın etkinlik ile ilgili tanımlamalar

Kapalı bir alan içerisindeki ışık kaynağından çıkan ışınlar incelendiğinde, dA birim alanından geçen ışın miktarının bulunması için her bir ışın dt süresi içinde $v dt$ kadar hıza ulaşır. Şekil 3.11'de birim yüzeyden geçen ışınlar gösterilmiştir.



Şekil 3.11. Birim yüzeyi geçen ışınlar

Birim yüzeyden geçen ışınların birim hacmi;

$$dV = (v \cos \theta dt) dA \quad [3.1]$$

Formülü ile hesaplanır. Bir yüzeyden geçen toplam ışın sayısı aşağıdaki gibi bulunur.

$$\begin{aligned} P(x) &= p(x) dv \\ P(x) &= p(x)(v dt \cos \theta) dA \end{aligned} \quad [3.2]$$

Parçacıklar her zaman aynı hızda ve aynı yönde hareket etmezler. Bu yüzden parçacıkların yönünde ω , dikkate alınması gerekir.

$$P(x, \omega) = p(x, \omega) \cos \theta d\omega dA \quad [3.3]$$

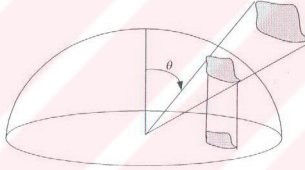
Buradan ikinci bölümde verilen uzay açı izdüşümü yasasına göre ışık akısı eşitliği yazılırsa;

$$d\Phi = \left[\int_{\Omega} L_i \cos \theta \cdot d\varpi \right] dA \quad [3.4]$$

Eşitlikteki Ω uzay açığı , R ise yarıküre üzerine gelen ışınırı belirtir. Şekil 3.12’de küresel koordinat sisteminde diferansiyel alanın xy düzlemindeki izdüşümünün alınması gösterilmiştir.

$$E \equiv \frac{d\Phi}{dA} \quad [3.5]$$

$$E = \int_{\Omega} L \cdot \cos \theta \cdot d\varpi$$



Şekil 3.12. Küresel koordinat sisteminde diferansiyel alanın xy düzlemindeki izdüşümünün alınması

Yukarıdaki eşitlikteki, $\cos \theta \, d\varpi$, çarpımı izdüşümü alınan yüzey açısını gösterir.

$$\int_{\Omega} \cos \theta \, d\varpi = \int_0^{2\pi} \int_0^{\pi} \cos \theta \sin \theta \, d\varpi \, d\theta = -2\pi \frac{\cos^2 \theta}{2} \Big|_0^{\pi/2} = \pi \quad [3.6]$$

Eğer ışığın tüm ışınları paralel olursa, bu ancak uzak bir kaynağın alanı aydınlatması ile mümkündür, formül aşağıdaki şekle dönüşür. İkinci bölümde verilen kosinüs yasasına göre E aşağıdaki gibi ifade edilir. E_0 birim alana dik aydınlık düzeyi ise;

$$E = E_0 \cos \theta \quad [3.7]$$

3.2.3.1. Işın etkinlik ve aydınlatma gücü

Işık kaynaklarından çıkan ışınlar nesnelerin yüzeyleri üzerine düşerler. Bu yüzeylerden yansıyan ışınların mitarına ışın etkinlik denir. B harfi ile gösterilir. Bir alanı terkeden birim alana ait ışın miktarıdır. Aydınlatma gücü olarak da isimlendirilir (29). Işık yayan bir yüzeyin ışık akısı yoğunluğuna ışınıtı denir (25). Burada R_o sembolü ışınıtı değerini göstermektedir

$$B = \int_{\Omega} R_o \cos \theta \cdot d\varpi \quad [3.8]$$

3.2.3.2. Bir ışık şiddeti eğrisi ile temsil edilebilen ışık kaynağı için ışık şiddeti

Bir ışık şiddeti eğrisi ile temsil edilebilen ışık kaynağından çıkan enerjinin dağılımı merkezden dışarıya doğru olur. Işın miktarı yoğunluğu verilen belirli bir yönde yansıdığı alanla yaptığı açıyla aşağıdaki gibi ifade edilir.

$$d\Phi = I(\varpi) d\varpi \quad [3.9]$$

Eşitlikteki $d\Phi$ terimi ışık akısını, $I(\varpi)$ bir ışık şiddeti eğrisi ile temsil edilebilen ışık kaynağının ışık şiddetini, $d\varpi$ ise küçük ışık hüzmesini belirtir. Bir ışık şiddeti eğrisi ile temsil edilebilen ışık kaynağı için ışık şiddeti;

$$I = \frac{\Phi}{4\pi} \quad [3.10]$$

olarak ifade edilir. Böylece yutulan toplam ışık akısı, Φ aşağıdaki hale gelir.

$$\Phi = \int I(\varpi) d\varpi \quad [3.11]$$

Tek bir ışık şiddeti eğrisi ile temsil edilebilen ışık kaynağından diferansiyel bir alana oluşan aydınlık düzeyi E , ışık kaynağından bakıldığında yüzey elemanı tarafından çevrilenen açının bulunması ile hesaplanabilir.

$$E = I \frac{d\varpi}{dA} = \frac{\Phi}{4\pi} \frac{\cos\theta}{|x-x_s|^2} \quad [3.12]$$

$|x-x_s|$, noktadan yüzey elemanına olan uzaklığı verir.

3.3. Kaplama Eşitliği

Yansıma eşitliği, gelen ışığın dağılımından ve malzemenin BRDF (iki yönlü yansıtma dağılım fonksiyonu) 'inden yansıyan ışık dağılımının hesabını mümkün kılmaktadır. İlk ve en çok tercih edilen model, basit ışık kaynaklarından kaynaklanan doğrudan aydınlatmadır. Ortam bütün olarak düşünülmez ve ışık kaynaklarının özellikleri ile gölgelenen yüzeyler dikkate alınır. Bu aydınlatma modeli lokal aydınlatma olarak adlandırılır. Yüzey üzerindeki bir noktanın ışık kaynağına görünür olup olmadığı denenerak gölgeler eklenebilir. Tüm ortam gözönüne alınan bu model ise global (genel) aydınlatma örneğidir. İkinci ve gerçek anlamda zor olan diğer durum ise dolaylı aydınlatmadır. Bu durumda ışık ortamdaki herhangi bir yüzeyden gelebilir ve gölgeleme burada çok önemlidir (30).

3.3.1. Yerel veya doğrudan aydınlatma

Yansıma eşitliği:

$$R_r(\vec{\omega}_r) = \int_{\Omega_i} f_r(\vec{\omega}_i \rightarrow \vec{\omega}_r) R_r(\vec{\omega}_i) \cos\theta_i d\varpi_i \quad [3.13]$$

ile ifade edilir. Aynı şekilde bir ışık şiddeti eğrisi ile temsil edilebilen ışık kaynağından elde edilen aydınlık düzeyi aşağıdaki gibi verilir.

$$E = \frac{\Phi}{4\pi} \frac{\cos \theta}{|x - x_s|^2} \quad [3.14]$$

Eğer ışık kaynağının yönü $\vec{\omega}_s$ ile verilirse bir ışık şiddeti eğrisi ile temsil edilebilen bir ışık kaynağının ışıntısı aşağıdaki gibi bir delta fonksiyonu ile ifade edilir. $L_i(\vec{\omega}_i)$;

$$R_i(\vec{\omega}_i) = \frac{\Phi}{4\pi|x - x_s|^2} \delta(\cos\theta_i - \cos\theta_s) \delta(\varphi_i - \varphi_s) \quad [3.15]$$

Yukarıdaki eşitliği yansıma eşitliğinde yerine konursa;

$$R_r(\vec{\omega}_r) = \frac{\Phi}{4\pi|x - x_s|^2} f_r(\vec{\omega}_r - \vec{\omega}_s) \cos\theta_s \quad [3.16]$$

elde edilir.

3.3.2. Global veya dolaylı aydınlatma

İlk adımda, bir yüzey üzerindeki aydınlatma başka bir yüzeyden yansarak gelen ışık dağılımı ile ilişkilendirilir. Işıntının bir yüzey boyunca sabit olduğu gerçeğini kullanarak, x üzerindeki ışıntıdan oluşan x' üzerine düşen ışıntı hesaplanır.

$$R_i(x'\vec{\omega}_i') = R_o(x, \vec{\omega}_o) \quad V(x, x') \quad [3.17]$$

Buradaki $\vec{\omega}_i$ x' 'den x' ne doğru yön vektörüdür. $\vec{\omega}_o$ ise, x den x'' ne doğru olan yön vektörüdür.

$$\vec{\omega}_i = -\vec{\omega}_o = \frac{x - x'}{|x - x'|} \quad [3.18]$$

$V(x, x')$, görünürlük fonksiyonudur. Eğer x ve x' kesinlikle görüntü ise 1 değerindedir, aksi halde sıfırdır. Bir sonraki adım, tüm gelen yönler üzerinden alınan yarı küresel integrali ortamdaki tüm alanlar üzerinden alınan alan integraline dönüştürmektir. Kaynak tarafından çevrilen açı iz düşümü oluşan edilen yüzey alanları ile ilişkilendirilerek kolayca dönüştürebilir.

$$d\omega'_i = \frac{\cos\theta_o \cdot dA}{|x - x'|} \quad [3.19]$$

Bu eşitlik iz düşümü açısı ile çarpılırsa;

$$d\omega'_i \cdot \cos\theta_o \cdot dA = G(x, x') \cdot dA \quad [3.20]$$

elde edilir.

Burada yer alan $G(x, x')$ geometrik fonksiyonunun değeri aşağıdaki gibidir.

$$G(x, x') = G(x', x) = \frac{\cos\theta'_i \cdot \cos\theta_o}{|x - x'|^2} \quad [3.21]$$

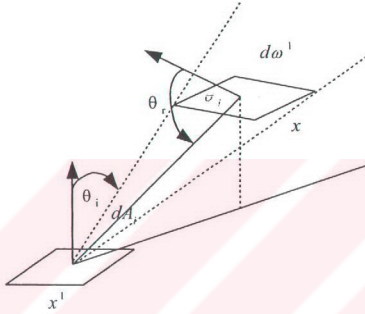
$G(x, x')$ değeri yansıma eşitliğine yerleştirilirse eşitlik S yüzeyleri üzerinden integral eşitliğine dönüşür.

$$R(x', \vec{\omega}') = \int_S f_r(x) R(x, \vec{\omega}) G(x, x') V(x, x') dA \quad [3.22]$$

Bu eşitlik çıkan ısıntılar yönleride içerdiğinden, geliş ve çıkış yönlerini temsil eden elemanlar bulunabilir. Yukarıdaki eşitlik bilgisayarda grafik yazılımına ilk olarak Kajiya tarafından kaplama eşitliği adıyla tanıtıldı. Bu, x' 'den x'' 'ne olan iki nokta aktarma yoğunluğu, $I(x, x')$ sembolü ile verilmiştir. Yoğunluk ölçüsü sadece yüzey konumu ile ilgili bir fonksiyondur ve çevresel açıları içermez. İki nokta arası ışık akısı eş. [3.33]'de ifade edilir.

Şekil 3.13'de iki nokta transfer geometrisi ve şekil 3.14'de üç nokta transfer geometrisi gösterilmiştir.

$$d\Phi = I(x \rightarrow x') dA dA' = R(x, \vec{\omega}) G(x, x') dA dA' \quad [3.23]$$



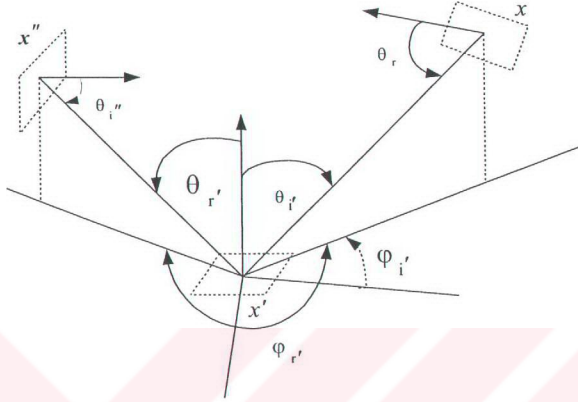
Şekil 3.13. İki nokta transfer geometrisi

Bu dA' yı dA' ne bağlayan ışın demeti içindeki akan ışık akıdır. Eşitliğin iki tarafı $G(x', x'') dA' dA''$ ile çarpılırsa bu forma getirilebilir. Eşitliğin çarpımdan sonraki hali aşağıdaki gibidir.

$$I(x' \rightarrow x'') = G(x', x'') \int f_r(x \rightarrow x' \rightarrow x'') V(x, x') I(x \rightarrow x') dA \quad [3.24]$$

Bu eşitlik x' den x' 'ne doğru giden ve x'' 'ne yansıyan ışık miktarını tanımlar. Böylelikle bu eşitlik çoklu nokta transfer eşitliği olarak da anılabilir. Miktar,

$$f_r(x \rightarrow x' \rightarrow x'') = f_r(x', \vec{\omega}'_i \rightarrow \vec{\omega}'_r) \quad [3.25]$$



Şekil 3.14. Üç nokta transfer geometrisi.

BRDF'in yeniden parametrelendirilmiş halidir. Kaplama eşitliğinin son haline erişmek için bir tek adım vardır.

Sadece opak yüzeylerden oluşan bir ortamda yüzeyler ışık kaynağı olarak tanımlanır. Oluşan ışık kaynağı değeri yüzeylerin ışıltı yutmasına bağlıdır (31).

$$R(x', \bar{\omega}') = R_e(x', \bar{\omega}') + \int_{\mathcal{S}} f_r(x) R(x, \bar{\omega}) G(x, x') V(x, x') dA \quad [3.26]$$

R_e değeri yutulan ışıntı değeridir.

3.3.3. Işın etkinlik eşitliği

Işın etkinlikte ortamdaki tüm yüzeyler Lambert yasasına uygun homojen dağınık yansıtıcı oldukları varsayılır. Böylece, BRDF gelen ve çıkan yönlerden bağımsızdır ve integralin dışına alınabilir.

$$\begin{aligned}
 R(x' \rightarrow x'') &= R_e(x' \rightarrow x'') + f_r(x') \cdot \int_S R(x \rightarrow x') G(x, x') V(x, x') dA \\
 &= R_e(x' \rightarrow x'') + \frac{\rho(x')}{\pi} \cdot \int_S R(x \rightarrow x') G(x, x') V(x, x') dA
 \end{aligned} \quad [3.27]$$

Homojen yansımada bir yüzeyden dışarıya çıkan ışıyım her yönde aynıdır ve ışıyım etkinlik B' nin π 'ye bölümüne eşittir. Bu da bizi bazı basitleştirmelere götürür. Kaplama eşitliği uzaydaki tüm noktalarda ışıyım enerjisinin konumunu ifade eder (32).

$$B(x) = E(x) + \rho(x) \int_S B(x') \frac{G(x, x') V(x, x')}{\pi} dA' \quad [3.28]$$

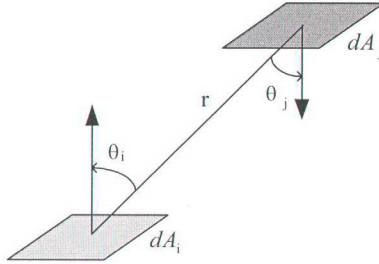
Böylece ışıyım etkinlik eşitliği bulunur.

3.3.4. Form faktörü

Sabit taban fonksiyonları için form faktör eşitlik [3.39]'da iki taban fonksiyonunun integralini içerir. Elemanlar arasında, bu form faktörü integralinin üç ayrı formülasyonu elde edilebilir (33). Şekil 3.17'de iki taban fonksiyonunun form faktörleri gösterilmiştir. Bu formülasyon form faktörü diferansiyel eşitliği, eş.[3.39]'da yerleştirilerek elde edilir. Böylelikle elemandan-eleman form faktörü, geometrik esas G_{ij} 'yi de içeren çift-alan integralidir.

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_1 \cdot \cos \theta_2}{\pi r^2} V_{ij} dA_j dA_i \quad [3.29]$$

Eşitlikte yeralan V_{ij} terimi diferansiyel alanlar dA_j ve dA_i arasındaki görünürlük terimidir.



Şekil 3.15. İki taban fonksiyonunun form faktörü

3.3.5. Sabit elemanların ışıma etkinliği

Yukarıda verilen eşitliklerde, her elemana ait sabit olduğu varsayılan yansıma ve yutma ile sabit taban fonksiyonları kullanılırsa karmaşıklık azaltılır. Sabit taban fonksiyonları, N_i , için Galerkin formülü yazılabilir. İlk olarak, yalnız 1 ve 0 değerlerini aldığı bilinen kutu taban fonksiyonları kullanılır.

$$\int_{\mathcal{S}} N_i(x) N_j(x) dA = \begin{cases} A_i, & i = j \text{ ise} \\ 0, & \text{aksi halde} \end{cases} = \delta_{ij} A_i \quad [3.30]$$

Burada yer alan δ_{ij} , Kronecker delta ve A_i ise i . elemanın alanını tanımlar. Aynı şekilde yutma fonksiyonu ise aşağıdaki gibi tanımlanır.

$$\int_{\mathcal{S}} E(x) N_i(x) dA = E_i A_i \quad [3.31]$$

E_i terimi i . elemana ait alanın ortalama yutma değeridir (34). Taban fonksiyonları birim değerlere sahip oldukları integralden dışarıya alınabilirler. Ayrıca yansıma fonksiyonu $\rho(x)$ 'in A_i üzerinde sabit bir ρ_i değerini aldığı varsayılır.

$$\int_{\mathcal{S}} N_i(x) \rho(x) \int_{\mathcal{S}} N_j(x') G(x, x') dA' dA = \rho_i \int_{A_i} \int_{A_j} G(x, x') dA_j dA_i \quad [3.32]$$

Bu sonuçlar, her i için eşitlik değerlendirilirse:

$$\left[\sum_{j=1}^n B_j \left[\delta_{ij} A_i - \rho_i \int_{A_i} \int_{A_j} G(x, x') dA_j dA_i \right] \right] = -E_i A_i = 0 \quad [3.33]$$

A_i 'ye bölünürse ve yutma terimi eşitliğin sağ tarafına alınırsa aşağıdaki eşitlik elde edilir.

$$\left[\sum_{j=1}^n B_j \left[\delta_{ij} - \rho_i \frac{1}{A_i} \int_{A_i} \int_{A_j} G(x, x') dA_j dA_i \right] \right] = E_i \quad [3.34]$$

veya

$$\left[\sum_{j=1}^n B_j \left[\delta_{ij} - \rho_i F_{ij} \right] \right] = E_i \quad [3.35]$$

Bir önceki formülde belirtilen F_{ij} , form faktörü olarak adlandırılır ve aşağıdaki gibi ifade edilir.

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} G(x, x') dA_j dA_i \quad [3.36]$$

Form faktörü, i . elemanı terkeden enerjinin, j elemanına ulaşan enerjiye bölümünü gösterir. Daha önceki formülasyonlar sırasında da olduğu gibi lineer sistem eşitlikleri matris eşitliği, $K.B = E$ ile ifade edilebilir. K giriş değerleri aşağıdaki gibi bulunur.

$$K_{ij} = \delta_{ij} - \rho_i F_{ij} \quad [3.37]$$

Buradan eş. [3.28] yeniden değerlendirilirse klasik ışın etkinlik eşitliği elde edilir.

$$B_i = E_i + \rho_i \sum_{j=1}^n B_j F_{ij} \quad [3.38]$$

A_i ile çarpılırsa,

$$B_i A_i = E_i A_i + \rho_i \sum_{j=1}^n B_j F_{ij} A_j \quad [3.39]$$

ve form faktörler arasındaki eşitlik kuralı kullanılırsa,

$$F_{ij} A_j = F_{ji} A_i \quad [3.40]$$

ve

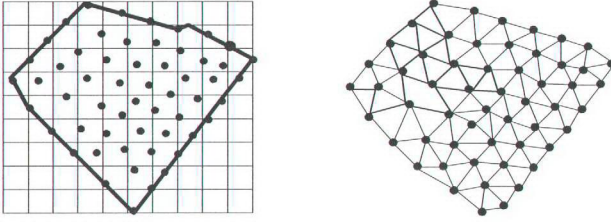
$$B_i A_i = E_i A_i + \rho_i \sum_{j=1}^n B_j F_{ij} A_j \quad [3.41]$$

elde edilir.

Bu eşitliğin fiziksel yorumu, bir elemanı terkeden toplam gücün, $B_i A_i$, sadece elemanın doğrudan yuttuğu ve yansıttığı ışınlarla bağlı olmasıdır. Bu yansıyan ışık ise ortamda bulunan bütün elemanlardan ayrılan ışıntıya bağlıdır. Ortam içindeki elemanlardan ayrılan ışıntının bir parçası tekrar bir elemana gelebilir. Bu ışıntı, elemanlar arası geometrik ilişkiye ve elemanların yansıtmasına bağlı olarak değişir (35).

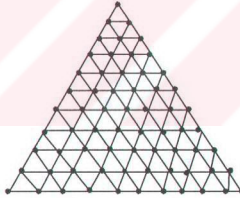
3.3.6. Ağ oluşturma

Işın etkinlik uygulamalarında, Linshinski metodu ile nesne kenarları oluşturulmuştur. Heckbert nesneleri tanımlama için üçgenlere bölme yaklaşımını getirmiştir. Şekil 3.16'da ilk düğüm algoritmaları ile bölümlene gösterilmiştir.



Şekil 3.16. İlk düğüm algoritmaları ile bölmeleme

Geliştirilen program yazılırken nesnelerin bölmeleme fonksiyonu ışın etkinlik fonksiyonundan önce yazılmıştır. Delaunay metodu ile üçgenlerin birbirine bağlanması ve düğüm noktalarının belirlenmesinde kullanılmıştır. Düzgün üçgensel nesneler eşit biçimli bölünür. Santimetrekareye düşen üçgen miktarı program içinde ayarlanarak sabitlenir. Şekil 3.17’de düzgün geometrik şekillerin düğüm noktalarının belirlenmesi gösterilmiştir. Düğüm noktalarına ışın etkinlik fonksiyonu uygulanır (36).



Şekil 3.17. Düzgün geometrik şekillerin düğüm noktalarının belirlenmesi

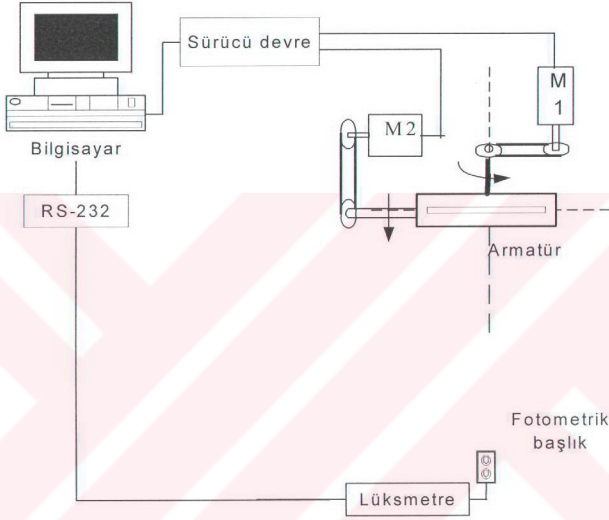
4. BİLGİSAYAR KONTROLLÜ GONİOFOTOMETRE TASARIMI

Geliştirilen çalışma bilgisayar kontrollü goniofotometre ve goniofotometrenin ürettiği verileri kullanarak sanal ortamlarda görüntü oluşturan bir yazılımdan oluşmaktadır.

Sistemin bilgisayar kontrollü goniofotometre bölümünün amacı hazırlanmış olan bir bilgisayar programı ile ışık şiddet verilerini almaktır. Yazılım ile açılara göre alınan verilerden eğri, eğrilerden ışık dağılım yüzeyi elde edilmesidir. Geliştirilen goniofotometre bilgisayar kontrollü olup ışık kaynağını hareket ettiren mekanizma ve fotometre başından oluşmaktadır. Goniofotometre'ye bağlı ışık kaynağının ışık şiddeti dağılımını incelemek için yaptığı düzenli hareketler adım motorları ile sağlanmıştır. Adım motorları açısal olarak hareket ettikleri için hareket aralığı, sargılara gönderilen sinyal sayısı kadar olur. Böylece adım açısına ve gönderilen sinyal sayısına bağlı olarak çok hassas olarak istenilen hareketler elde edilmiş olur. Bu çalışmada açık döngülü sürücü sistemi kullanılmıştır. Adım motorlarının uygun olarak çalışması için sargılara gönderilecek sinyallerin de uygun bir şekilde üretilmesi gerekir. Motorlar, bilgisayar programından gelen sinyale göre istenilen yön ve adım sayısı kadar hareket ederler. Dönüş yönü, yön girişinin lojik durumu tarafından belirlenir. Lojik-1 saat ibresi yönü, lojik-0 ise saat ibresinin tersi yönüdür. Lojik sıralayıcının bazı uygulamaları tek yönlü olup, yön sinyal ucu yoktur. Eğer hareket bir artırılsa, bir adım meydana getirilir. Fakat artış iki veya daha fazla adımla oluşturulursa, lojik sıralayıcıdan önce diğer bir katta üretilen sinyallerin uygun bir şekilde yönlendirilmesi gerekir.

Adım motorlarının, bilgisayarın paralel girişinden gönderilen sinyallere göre çalışmasını sağlamak için bir sürücü devre tasarlanmıştır. Motorların istenilen adım, yön ve hızda çalışması bu sinyallere bağlıdır. Sürücü devre ve motorların çalışması için gerekli olan enerji bir 12V DA güç kaynağından sağlanmaktadır. Fotometre başı tarafından algılanan ve lüksmetrede okunan değerler RS-232 kartı ile bilgisayarın seri girişine gönderilir (37). Bu fotometrik bilgiler program tarafından kullanılır.

Armatür ile ilgili fotometrik bilgileri değerlendiren ve motor fonksiyonlarını yönlendirilen C++ Builder 5.0 ile yazılan bu yazılım ile goniofotometreye bağlanmış olan armatürün ışık şiddet eğrileri çıkartılır. Sistemin yapısı Şekil 4.1’de gösterilmiştir.



Şekil 4.1. Sistemin çalışma yapısı

4.1. Goniofotometre

Işık kaynağı veya fotometre başından birinin sabit, diğerinin düzenli olarak yaptığı hareketler ile bir ışık kaynağının ışık şiddet dağılımının incelenmesi için kurulan sistem goniofotometredir (38). Goniofotometreler tasarımı için ölçüm düzlemi ve yapım tipi seçimi yapılmalıdır.

Ölçüm düzlemi, bir ışık kaynağının ışık dağılımını kesen ara kesitlerdir. Fotometrik başlığın konumuna göre ışık kaynağının uzaydaki yerini tarif etmek için kullanılır.

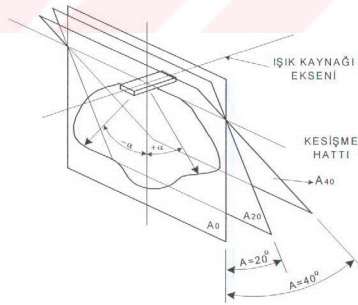
Uygulamada en çok kullanılan ışık yayılım doğrultusunu kapsayan ölçüm düzlemleri A, B, C olarak adlandırılır. Goniofotometreler yapım tipleri aşağıda verilmiştir.

4.1.1. Ölçüm düzlemi

Armatürün ışık şiddet eğrilerini çıkartmak için uygun olan ölçüm düzlemlerinin seçimi, ışık kaynaklarının çeşidine ve bu ışık kaynaklarının goniofotometre yapısına bağlıdır.

4.1.1.1. A düzlemi

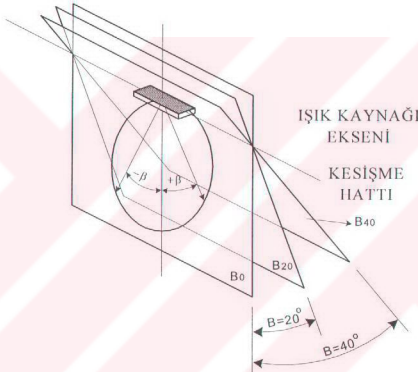
A düzlemleri dikdörtgen ışık kaynağı eksenini enine kesitte kesen düzlemlerden oluşur. Işık dağılımını inceleyen A düzlemlerinin tümü ışık kaynağının fotometrik merkezinde kesişirler. Düzlemlerin kesiştiği bu noktaya kesişme hattı denir. A düzleminde ışık kaynağının fotometrik merkezinin içinden geçen kesişim hattının eksenini ışık kaynağının eksenine diktir. Işık kaynağı A düzlemindeki hareketini enine kesitte yatay düzlemde sabit fotometrik başlık karşısında yapar. Şekil 4.2'de A düzlemleri gösterilmiştir.



Şekil 4.2. A düzlemleri

4.1.1.2. B düzlemi

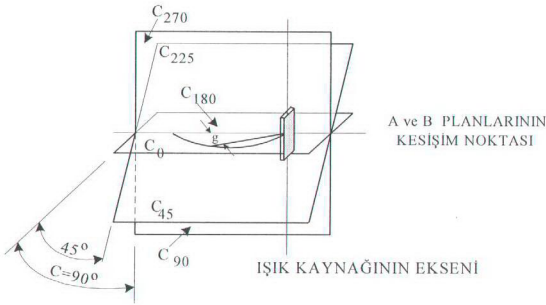
B düzlemleri dikdörtgen ışık kaynağı eksenini boyuna kesitte kesen düzlemlerden oluşur. Işık dağılımını inceleyen B düzlemlerinin tümü ışık kaynağının fotometrik merkezinde kesişirler. Bu kesişim hattı ile ışık kaynağının fotometrik merkezi aynı paraleldir. Işık kaynağı B düzlemindeki hareketini boyuna kesitte yatay düzlemde sabit fotometrik başlık karşısında yapar. Uygulamalarda aşırı ışık veren armatürlerin ışık şiddet eğrilerinin çıkartıldığı düzlemdir. Şekil 4.3’de B düzlemleri gösterilmiştir.



Şekil 4.3. B-Düzlemleri

4.1.1.3. C düzlemleri

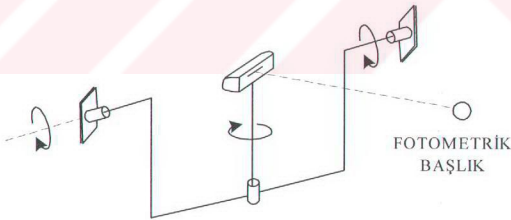
C düzlemleri ışık kaynağının fotometrik merkezinin yatay eksenini dik olarak kesen düzlemlerden oluşur. A ve B düzlemlerindeki ışık kaynağı eksenlerinin yatay ekseninde kesiştiği fotometrik merkezin içinden dik olarak geçen eksen çizgisine C düzlemlerinin kesişme hattı denir. C düzlemlerinin ışık kaynağı dağılımını incelediği kesitler kesişme hattı merkezlidir. Şekil 4.4’de C düzlemleri gösterilmiştir.



Şekil 4.4. Işık kaynağına dik bağlantılı C-düzlemleri

4.1.2. Yapım prensipleri

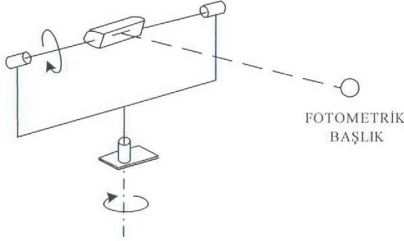
Tip 1. Sabit olan fotometrik başlık referans kabul edilirse, ışık kaynağı yatay ekseninde boyuna hareket ettirildiğinde A düzlemi prensiplerine göre hareket ettirilmiş olur. Işık kaynağı dikey ekseninde hareket ettirilirse, B düzlemi hareketini yapar. Şekil 4.5'de ışık kaynağını döndürme düzeneği olan goniofotometre Tip 1 gösterilmiştir.



Şekil 4.5. Işık kaynağını döndürme düzeneği olan goniofotometre Tip 1.

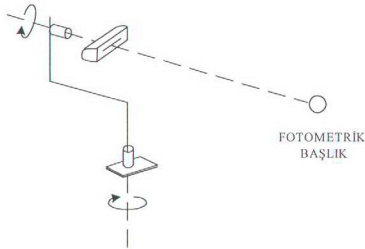
Tip 2. Sabit olan fotometrik başlık referans kabul edilirse, ışık kaynağı yatay ekseninde boyuna hareket ettirildiğinde A düzlemi prensiplerine göre hareket ettirilmiş olur. Işık kaynağı yatay ekseninde enine hareket ettirilirse, B düzlemi hareketini yapar.

Şekil 4.6'da ışık kaynağını döndürme düzeneği olan goniofotometre Tip 2 gösterilmiştir.



Şekil 4.6. Işık kaynağını döndürme düzeneği olan goniofotometre Tip 2.

Tip 3. Sabit olan fotometrik başlık referans kabul edilirse, Goniofotometrenin bu yapısında iki eksenle üç düzlem hareketi yapılır. Işık kaynağı yatay boyuna hareket ettirilmesi ile A düzleminde hareket elde edilir. Işık kaynağı kendi etrafında döndürülerek C düzleminde hareket elde edilir. Işık kaynağının C düzleminde 90° döndürülmesi ile yatay eksenle enine hareket sağlanır. Bu durum B düzlemi hareketine imkan verir. Bu uygulamanın üstünlüğü ışık kaynağının ışık dağılım yüzeyini çıkartan C düzlemindeki hareketleri sağlayabilmesidir (41). Şekil 4.7'de ışık kaynağını döndürme düzeneği olan goniofotometre Tip 3 gösterilmiştir.



Şekil 4.7. Işık kaynağını döndürme düzeneği olan goniofotometre Tip 3

C düzlemlerinin 0° 'deki konumu A düzlemini, 90° 'deki konumu ise B düzlemini kapsar. Bu düzlem ile diğer düzlemlerle ulaşılan bilgilere de ulaşılmış olur. Goniofotometrenin yapısına göre C düzlemleri yatay ve dikey olarak tasarlanırlar.

4.2. Goniofotometre Seçimi

Bu uygulamada ışık kaynağını döndüren sisteme sahip goniofotometre kullanılmıştır. Goniofotometrenin yapısı Tip-3'e göre tasarlanmıştır. Tip-3'ün ışık kaynağına yaptırdığı düzenli hareketler adım motorları ile sağlanmıştır. Goniofotometrenin hareketleri ve buna bağlı olan ölçümler bilgisayar programı ile kontrol edilmiştir. CIE'nin 70 numaralı standartına uygun olarak hazırlanan sistemin uygulamadaki mekanik düzeneklerinin tasarımı diğer örneklerinden özgün olarak tasarlanmıştır (1).

Işık kaynağını döndüren sisteme sahip bir goniofotometrenin Tip-3'e göre seçilmesinin avantajları şunlardır. Işık kaynağı, yatay eksenine enine ve boyuna hareket ettirilerek hem dairesel hem de dikdörtgen ışık kaynaklarının ışık şiddet eğrilerinin çıkartılmasına imkan verir. Sistemle yapılacak ölçümler için büyük bir uygulama alanına ihtiyaç yoktur. Işık şiddet dağılımı daha az hareket ile incelenir.

4.3. Motor Seçimi

Açısal konumunu adımlar halinde değiştiren motorlara denir. Adım motoru sargılarına uygun sinyaller gönderilerek kontrol edilir. Herhangi bir uyarımda, motorun yapacağı hareketin ne kadar olacağı, motorun adım açısına bağlıdır. Adım açısı motorun yapısına bağlı olarak 90° , 45° , 18° , 7.5° , 1.8° veya daha değişik açılarda olabilir. Motora uygulanacak sinyallerin frekansı değiştirilerek motorun hızı değiştirilebilir (39). Adım motorlarının dönüş yönü uygulanan sinyallerin sırası değiştirilerek saat ibresi yönü (CW) veya saat ibresinin tersi yönünde (CCW) olabilir. Adım motorlarının hangi yöne doğru döneceği, devir sayısı, dönüş hızı gibi değerler mikroişlemci veya bilgisayar yardımı ile kontrol edilebilir. Adım motorlarının hızı, dönüş yönü ve konumu her zaman bilinmektedir. Bu özelliklerinden dolayı adım motorları çok hassas konum kontrolü istenen yerlerde

kullanılırlar (2). Adım motorları yapım şekillerine göre sınıflandırılırlar. Değişken relüktanslı adım motoru, sabit mıknatıslı adım motoru ve melez yapılı adım motoru en çok bilinen adım motoru çeşitleridir.

Bu çalışmada adım motorlarının hassas olarak hareket etme ve sayısal olarak kontrol edilebilme özelliklerinden faydalanılarak mekanik hareketlerin sağlanması için iki adet karışık yapılı adım motoru kullanılmıştır. Yapılmış olan ışık şiddet eğrisi çıkartan sistemde goniofotometrenin ışık kaynağını hassas açılarda hareket ettirmesi sağlanmıştır.

4.2.1 Adım motorlarının uyarıtımı

Sistem motorları için tek faz uyarıtımı, iki faz uyarıtımı ve yarım adım uyarıtımı tanıtılmıştır. Uyarıtım metodunun seçimi aşağıda anlatılmıştır.

4.3.1.1. Tek-faz uyarıtımı

Motor sargılarının sadece birinin uyarıldığı uyarıtım cinsine tek-faz uyarıtımı adı verilir. Çizelge 4.1'de 4-fazlı adım motoru için tek-faz uyarıtım sırasındaki fazların durumu görülmektedir. Bu uyarıtım metodunda rotor her bir uyarıtım sinyali için tam adımlık bir hareket yapmaktadır. Uyarıtım dönüş yönüne bağlı olarak sıra ile yapılır. Burada fazların uyarıtım sırası saat ibresi yönündeki (CW) dönüş için F1, F2, F3, F4, saat ibresinin tersi yönü (CCW) için F4, F3, F2, F1 şeklinde sargılar enerjilenir.

Çizelge 4.1. Tek-faz uyarıtımın faz uyarıtım sıralaması

Adım	R	1	2	3	4	5	6	7	8
Faz 1	×				×				×
Faz 2		×				×			
Faz 3			×				×		
Faz 4				×				×	

4.3.1.2. İki-faz uyarıtımı

Motor sargılarının ikisinin sıra ile aynı anda uyarıldığı uyarıtım cinsine iki-faz uyarıtım adı verilir. Çizelge 4.2' de 4-fazlı adım motoru için iki-faz uyarıtım sırasındaki fazların durumu görülmektedir.

Çizelge 4.2. İki-faz uyarıtımın faz uyarıtım sıralaması

Adım	R	1	2	3	4	5	6	7	8
Faz 1	x	x			x	x			x
Faz 2		x	x			x	x		
Faz 3			x	x			x	x	
Faz 4	x			x	x			x	x

Burada iki faz aynı anda uyarıldığından stator ve rotor kutupları arasındaki ilişki bir-faz uyarıtımlıya göre farklıdır. İki faz uyarıtımlıda rotorun geçici durum tepkisi tek-faz uyarıtımlıya göre daha hızlıdır. Fakat burada güç kaynağından çekilen güç iki katına çıkmaktadır.

4.3.1.3. Yarım adım uyarıtımı

Bu uyarıtım yönteminde tek-faz uyarıtımı ile iki-faz uyarıtımı ardarda uygulanır. Burada rotor her bir uyarıtım sinyali için yarım adımlık bir hareket yapmaktadır. Çizelge 4.3'de fazların uyarıtım sırası görülmektedir.

Çizelge 4.3. Yarım adım (melez) uyarıtımın faz uyarıtım sıralaması

	R	1	2	3	4	5	6	7	8
Faz 1	x	x						x	x
Faz 2		x	x	x					
Faz 3				x	x	x			
Faz 4						x	x	x	

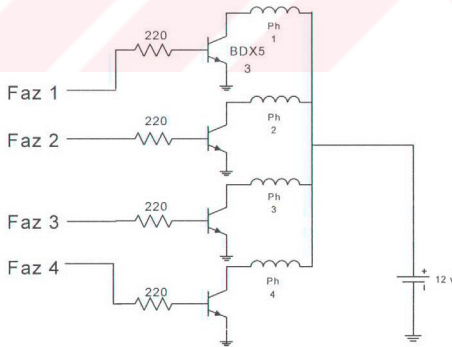
4.3.1.4. Uyarıtım metodunun seçimi

Adım sayısının fazla olması ve dolayısıyla adım açısının küçük olması istenen yerlerde yarım adım uyarıtımı tercih edilir.

Sistemin tepkisinin hızlı olması ve tutma momentinin güçlü olması istenen yerlerde iki-faz uyarıtımı kullanılmalıdır. Fakat bu durumda enerji tüketimi iki katına çıkmaktadır. Dolayısı ile enerji tüketiminin az olması istenen yerlerde tam adım konumu gerçekleştirmek için tek-faz uyarıtımı kullanılmalıdır (38). Sistemde kullanılan adım motorlarının uyarıtım metodu tutma momentinin daha iyi olması için iki faz uyarıtımı kullanılmıştır.

4.4. Adım Motoru Sürücü Sistemi

Bu uygulamada kullanılan adım motorları 4-fazlı, melez yapılı adım motorları olup tam-adım ve her adımda iki faz uyarımlı olacak şekilde sürülmektedir. Şekil 4.8'de adım motoru güç devresi gösterilmiştir.



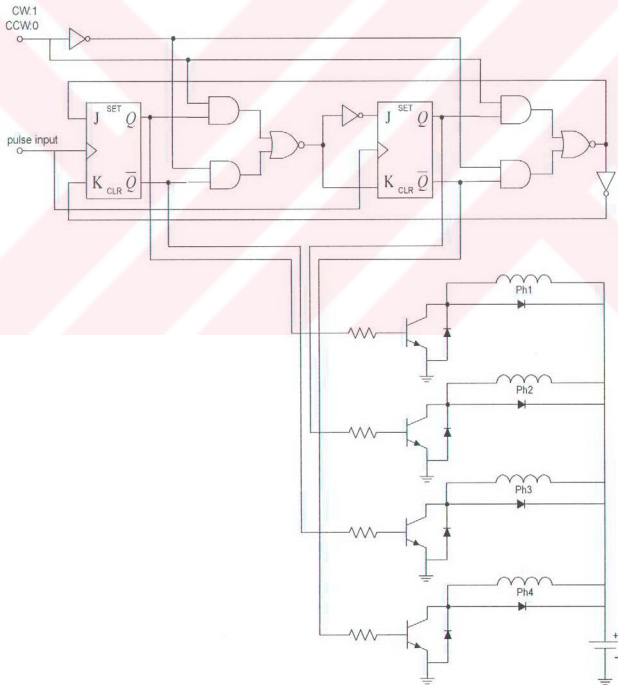
Şekil 4.8. Adım motoru güç devresi

Sargıların uyarıtımı için her bir faza darlington güç transistörü ve koruma diyotu kullanılmıştır. Motorların çalışması için gerekli olan enerji DA güç kaynağından

sağlanmaktadır. Normalde 4-fazlı motorun sürülmesi, ileri-geri yönde hareket etmesi ve fazların sırayla enerjilenmesi için bilgisayardan 4-bitlik sinyal gerekmektedir.

Bu çalışmada, lojik sıralayıcı olarak FF CMOS entegrelerinden oluşmuş bir sıralama devresi kullanılmıştır. Lojik sıralayıcının, sıralama yapması için bir saat sinyali ve yön sinyali gerektirdiğinden, her bir motor için 4 bitlik çıkış yerine 2 bitlik bilgi yeterli olmaktadır.

Şekil 4.9'da adım motorlarını çalıştıran sürücü devre verilmiştir. Bilgisayarın yön sinyali çıkışı 1 seviyesinde ise motor ileri, 0 seviyesinde ise geri yönde dönmektedir



Şekil 4.9. Adım motorlarını çalıştıran sürücü devre

Lojik sıralayıcının fazları sıralaması ileri yön için çizelge 4.4'de, geri yön için ise çizelge 4.5.'de verilmiştir.

Çizelge 4.4. İleri yön için faz sıralaması (CW)

Adım	1	2	3	4	5
Faz1	1	0	0	1	1
Faz2	1	1	0	0	1
Faz3	0	1	1	0	0
Faz4	0	0	1	1	0

Çizelge 4.5 Geri yön için faz sıralaması (CCW)

Adım	1	2	3	4	5
Faz1	0	0	1	1	0
Faz2	0	1	1	0	0
Faz3	1	1	0	0	1
Faz4	1	0	0	1	1

Sürtücü devresinin yön girişine (1) bilgisi geldiğinde fazlar çizelge 4.1' de sıralanarak adım motoru tam adım ve iki-faz uyarımlı olacak şekilde saat ibresi yönünde döner. Yön bilgisine (0) girildiğinde ise fazlar çizelge 4.2' deki gibi sıralanır. Bu durumda motor saat ibresi tersinde döner.

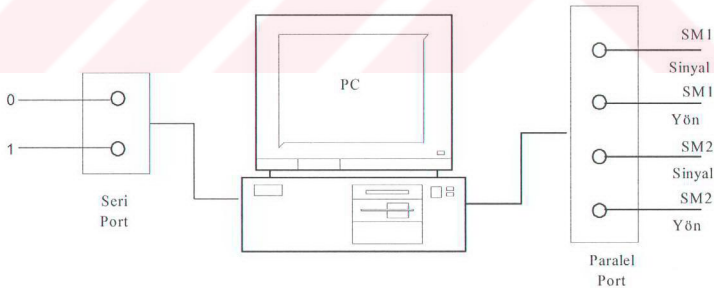
4.5. Bilgisayar Ortamından Dış Ortama Bilgi Aktarımı

Çevre birimlerinin bilgisayar ile bağlantısı ve bilgi alışverişi yapabilmesi, çeşitli giriş/çıkış arabirimleriyle sağlanır. Bilgisayar kasasının arkasında monitor, klavye, fare için özel bağlantı girişlerinin bulunduğu görülebilir. Diğer çevre birimlerini bilgisayara bağlamak için, genel amaçlı giriş/çıkış arabirimleri tasarlanmıştır.

Çalışmada kullanılan C programlama dilinde paralel porta bilgi göndermek için kullanılan komut Outport ve outportb komutları' dır. Outportb komutu bir bayt'lık outport komutu ise bir word'lük bilginin çıkışa aktarılmasını sağlar. LPT1 işletim etiketiyle kullanılan birim bilgi akışına yön verir. Program içinde motorların kontrolü paralel port ile sağlanmış olup, LPT1 işletim etiketiyle kullanılan birim bilgi akışına yön vererek kontrol sağlanmıştır (40).

Bir bilgisayar programı mantığı anahtarlamaya dayalıdır. Bir karakter 8 on-off kodları (yada bitleri) kullanılarak elde edilebilir. Tipik bir Byte 01001110, 11101101 , gibi meydana gelir. Bilgisayar içerisinde "1" +5 volt (YÜKSEK) ve "0" Şase (ALÇAK) olarak temsil edilir.

Bu bilgiler zamanlı gerilim darbelerinin bir serisi olarak (seri transfer) veya sekiz paralel kablodan (paralel transfer-8 bit) veri olarak gönderilebilir. C++ programlama dili ile hazırlanan yazılım içinde yer alan bir alt program ile lüksmetre tarafından gönderilen veriler onluk sayı düzenine çevrilerek lüksmetre ile bilgisayar arasında bilgi akışını sağlamıştır. Şekil 4.10'da bilgisayar ara birim bağlantıları gösterilmiştir.



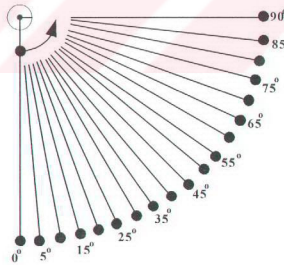
Şekil 4.10. Bilgisayar ara birim bağlantıları

4.6. Mekanik Sistem

Yapılan uygulamada mekanik sistem adım motorları ile hareket ettirilmektedir. Adım motorlarının döndürme işlevi sonsuz dişliler ile ışık kaynağını hareket ettiren mekanizmaya aktarılmıştır. Tasarlanan mekanik düzenekte kullanılan sonsuz dişliler ile adım motorlarının üzerine düşen yük en aza indirilmiştir.

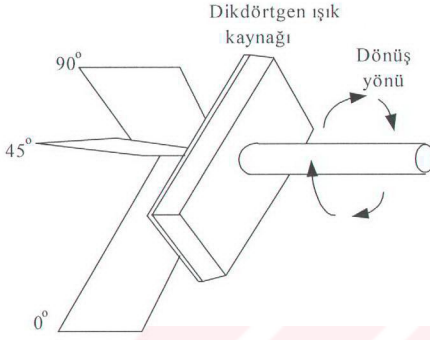
4.6.1. Sistem içinde motorların çalışması

Programın kontrol ettiği birinci adım motoru ışık kaynağını 0° - 90° arasında sağdan sola doğru 5° lik açılarla döndürür. Birinci adım motorunun, adım sayısı ve derecesi istenirse yazılımdan ayarlanabilir. Ölçüm sayısı değiştirilmezse sistem 18 değişik noktada aydınlık düzeyi ölçülmesine imkan verir. Şekil 4.11'de birinci adım motorunun sisteme yaptırdığı hareket gösterilmiştir. Her 5° de bir ışık kaynağının fotometre başı üzerinde oluşturduğu aydınlık düzeyi değeri program tarafından lüksmetreden okunarak ilgili dosyaya kaydedilir. Bu işlem bittiğinde birinci adım motoru ışık kaynağını başlangıç konumuna geri getirir (41).



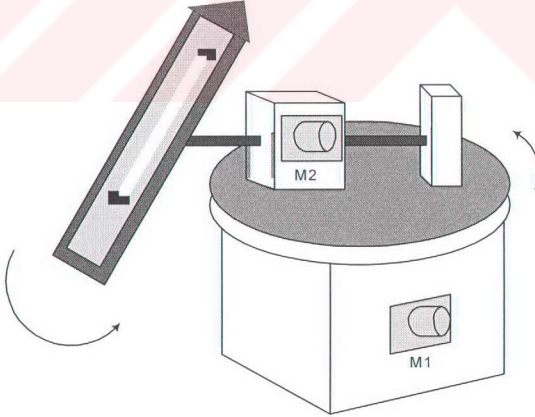
Şekil 4.11. Birinci adım motorunun sisteme yaptırdığı hareket

İkinci adım motoru ışık kaynağını C düzleminde hareket ettirir. Bu hareket ile ışık kaynağı enine ve boyuna hareket eder (42). Şekil 4.12'de ışık kaynağının C düzleminde hareketi gösterilmiştir.



Şekil 4.12. Işık kaynağının C düzleminde hareketi

Işık kaynağının yatay veya düşey eksenindeki hareketi armatürlerin ışık şiddet eğrilerinin çıkartılması için kullanılır. Şekil 4.13’de sisteme bağlı dikdörtgen armatür gösterilmiştir.



Şekil 4.13. Sisteme bağlı dikdörtgen armatür

4.7. Işık Şiddet Eğrilerin Elde Edilmesi

Hazırlanan bilgisayar programı çalıştırıldığında ilk olarak armatür tipi seçimi formu ile karşılaşılr. Işık şiddet eğrisi çıkartılacak olan armatürün tipine göre, dairesel, dikdörtgen, çokgen armatür tiplerinden biri tercih edilir. Kullanıcı armatürün karakteristik özelliklerinin girilen forma ölçme uzaklığını ve kaynağın ışık akısı değerlerini yükler. Armatür tercihine göre, program sistemi yönlendirir.

Programın başlangıç menüsünde dairesel armatür tipi tercih edilirse, birinci adım motorunun her dönme hareketinde lüksmetreden okunan aydınlık düzeyi değeri bilgisayar programının ilgili dosyasına kaydedilir. Bu sistemde denenen bir armatürün standart abak üzerinde başka bir armatürle karşılaştırılabilmesi için o armatüre ait ışık akısı 1000 lümen değeri ile orantılanır. Bu oran ile dosyaya atılmış olan aydınlık düzey değerleri çarpılarak yeniden kaydedilir. Örneğin lüksmetrede okunan aydınlık düzeyi 50 lüks ise ve ışık kaynağının ışık akısı 2000 lm ise bu durumdaki normalize edilmiş aydınlık düzeyi;

$$\frac{1000}{2000} \times 50 = 25 \text{ lükstür} \quad [4.1]$$

Aydınlık düzeyi hesaplandıktan sonra girilen bilgiler doğrultusunda ışık şiddeti her adıma göre şu şekilde hesaplanır;

$$I = \frac{E r^2}{\cos_{\epsilon_2} \Omega_0} \quad [4.2]$$

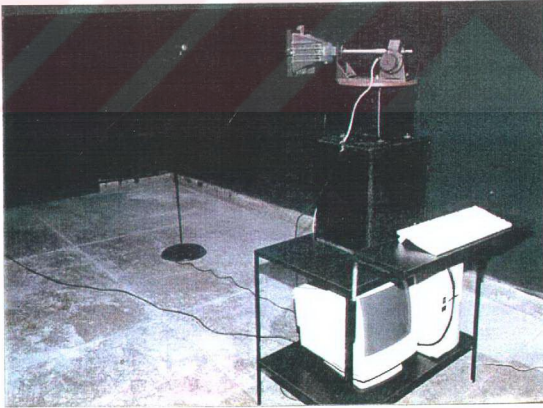
Eşitliği kullanılarak program bir eğrinin çizimi için kullanıcının programdan seçtiği adım sayısı kadar hesaplama yaparak bir dosyaya kaydeder. Sistemin seçilen armatür tipine göre hareketi birinci motorun sağdan sola 90°'lik hareketi dairesel tip armatür için yeterlidir. Dikdörtgen ve çokgen tip armatür için sistem birinci motorun 90°'lik

hareketi sonunda durur. İkinci motor armatürü kendi eksenini etrafında 90° döndürür. Soldan sağa doğru birinci motor armatürü hareket ettirir. Bu hareketi sırasında ölçümler alınır.

Dosyadaki bilgiler kullanılarak ölçüm ile hesaplanan ışık şiddeti abak üzerindeki ölçeklendirilmiş daire üzerine yerleştirilir. Işık şiddeti değerlerinin birimi cd/klm olarak abak üzerine işaretlenmiş olur. Bu şekilde armatürlerin ışık şiddet eğrisi çıkartılır.

Goniyofotometre ışık kaynağını döndürerek normale göre açılı hareket yaptırmaktadır. Bu yüzden ışık kaynağı ile fotometre başı arasındaki kosinüs açısının formülde kullanılmasına gerek yoktur.

Armatürlerin matematiksel olarak tanımlanması için gerekli olan ışık şiddet verilerini açılara göre bilgisayar yardımıyla ölçen ve hesaplayan goniyofotometre sisteminin fotoğrafı ve deneylerin yapıldığı karanlık oda şekil 4.14'de gösterilmiştir.



Şekil 4.14. Kurulan goniyofotometre sisteminin fotoğrafı ve karanlık oda

4.8. Sistemin Özellikleri

Seçilen gonifotometre tipi ışık kaynağını döndüren sisteme sahip Tip 3 CIE'in 70 nolu standardı ve DIN 5032-1 standartına göre yapılmıştır. Bu goniofotometre ile armatürlerin dairesel, dikdörtgen ve çokgen tiplerinin ışık şiddet değerleri alınabilmektedir. Sisteme ilave edilmiş bilgisayar desteği ve yazılımla ışık şiddeti eğrileri ve ışık dağılım yüzeyi çıkartılabilmektedir. Ölçümler için motorların açılmal pozisyon aralığı programla ayarlanabilir. Sistemin ölçümlerinde kullanılan fotometre başı $V(\lambda)$, UV-response, UV response ve dedektör yapısı Si-fotovoltaic hücre filitreleme, lüksmetrenin ölçme alanı 1 – 50000 lüks böylece ölçüm mesafesine bağlı olarak ölçümün kullanım alanı belirler. Sistemle lamba tipi seçimi halojen lamba, cıva buharlı lamba, flouresan lamba, akkor flamanlı lamba, kompak flouresant lamba ve ledlerle yapılan aydınlatmaların ölçümleri yapılabilmekte sisteme kaydedilirken lamba seçilebilmektedir.

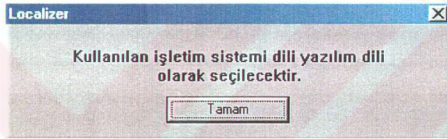
4.8.1. Sistemin hata kaynakları

Sistemde ölçüm yapılırken bir çok hata üretilir. Bu hataların en aza indirilmesi için aşağıdaki hata kaynaklarına dikkat edilmesi gerekir. Yatay ve düşey eksenlerin kesişmemesine, dönme sırasında kullanılan parçaların titrememesi, fotometre başının yanlış yerleştirilmemesine, pürüzlü dönme eksenini veya dengesiz hareketlerin oluşmamasına, mekanik sistemdeki dişlilerin verilen hareket sırasında diş atlamamasına dikkat edilmelidir. Sisteme verilen gerilim 220 V ve 50 Hz olmalıdır. Sisteminin gerilim dalgalanmaları, ölçümleri karşılaştırıldığında hatalara neden olduğu görülmektedir. Bilgisayar veri alım hızı 20 msn olarak belirlenmiştir. Bu durum motorların dönüşü ve programla belirlenen değere gelmesiyle senkron olamamaktadır. Deneylerin yapılacağı yerin özellikleri de sistemde hata nedenleri olabilmektedir. Oda sıcaklığı $15^{\circ}\text{C} < T < 35^{\circ}\text{C}$ arasında olmalıdır. Flouresan lamba gibi deşarj lambaların verimlerinin bu sıcakların altında ve üstünde çok farklı değerler verdikleri görülmüştür. Ayrıca odanın nem oranının en fazla % 20 civarında olmasına dikkat edilmelidir. Uygulama alanı siyah mat boya ile boyalıdır. Kesinlikle ışık almamalıdır.

5 GONİOFOTOMETRİK ÖLÇÜM YAZILIMININ TANITILMASI

5.1. Program Kontrol Alanı

Programa giriş alanı localizer alanıyla başlar. Localizer alanın amacı menülere farklı dil seçenekleri sağlamaktır. Alanın tamam kısmına tıklandığında uygulamanın yapıldığı bilgisayardaki kurulu olan işletim sisteminin yazım dilini kullanır. Şekil 5.1’ de localizer alanı gösterilmiştir.



Şekil 5.1. Localizer alanı

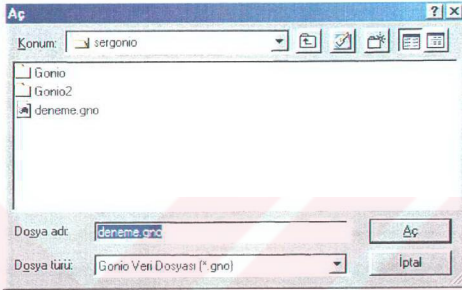
Programın yönetici alanı dosya, dil ve yardım menülerinden oluşmaktadır. Dosya alanı yeni, yükle, kaydet, farklı kaydet ve çık gibi alt alanlardan oluşur. Şekil 5.2’de dosya, dil ve yardım menüleri gösterilmiştir.



Şekil 5.2. Dosya, dil ve yardım menüleri

Sistemde goniofotometre çalıştırılmadan daha önce kaydedilmiş olan dosyaları kullanılması isteniyorsa yükle alt alanı kullanılır. Yükle alanı programın yüklü olduğu klasörü ekrana getirir. Şekil 5.3’ de klasörden sisteme dosya ilave edilmesi gösterilmiştir.

Goniofotometre ile ölçümü yapılan armatür verileri kaydet alt alanı ile programın yüklü olduğu klasöre kaydedilir. Farklı kaydet alt alanı ile farklı bir klasöre kaydedilir. Dil alt alanı farklı dillerde programı yüklemek için kullanılır.

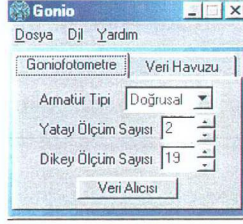


Şekil 5.3. Klasörden sisteme dosya ilave edilmesi

Yazılım, gonio konsolu ve veri havuzu olmak üzere iki temel bölümden oluşmaktadır. Gonio konsolu, goniofotometrenin hareketleri ve buna bağlı olan ölçümleri kontrol etmek için tasarlanmıştır. Veri havuzu, yeni ölçümü tamamlanmış olan bir armatür veya daha önceden sisteme kaydedilmiş olan bir armatüre ait verilerin saklanması, yeniden isimlendirilmesi, silinmesi, ıstık şiddet verilerinin iki ve üç boyutta bilgisayar ortamında gösterilmesi ve üç boyutlu sanal ortamlarda kullanılması için hazırlanmıştır.

5.1.1. Gonio konsolu

Goniofotometrenin tüm kontrolleri bu alandan yapılır. Şekil 5.4' de gonio konsolu gösterilmiştir.

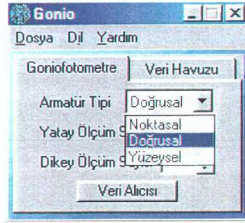


Şekil 5.4. Gonio konsolu

Armatür tipi seçimi, yatay ölçüm sayısı, dikey ölçüm sayısı ve veri alıcısı alt alanlarından oluşur. Goniofotometre alanında bulunan armatür tipi ölçümü yapılacak olan noktasal, doğrusal ve yüzeysel tip olarak seçilir. Yatay ölçüm sayısı alanı ile adım motorlarının adım sayısını ayarlayarak ölçümün kaç adımda gerçekleştirileceğini kullanıcı isteğe göre düzenler. Dikey ölçüm sayısı alanı armatürün geometrik yapısına göre program tarafından seçilir

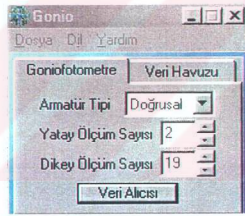
Armatürler geometrik yapılarına göre noktasal, doğrusal ve yüzeysel olmak üzere üç çeşittir. Noktasal armatürlerin dikey eksenindeki tüm açılardaki eğrileri birbirinin aynısı olduğundan program dikey ekseninde bir ölçüm yaptırır. Doğrusal armatürlerde ise en büyük ve en küçük eğri değerleri 0° ve 90° açılarda alınacağından program dikey ekseninde iki ölçüm yaptırır. Bu açılar arasındaki diğer eğriler program tarafından bulunur. Yüzeysel armatürlerde ise kullanıcı yatay ve dikey ölçüm sayısını kendisi belirler.

Ölçüm sayısı ayarı yapılırken adım motorunun açısına uygun olarak hazırlanmış olan sayılar boşluğa yazılabileceği gibi kullanıcı isterse boşluğun yanındaki ok sembolü altındaki hazır ölçüm sayılarından birini fare ile seçebilir. Şekil 5.5’de armatür tipinin seçimini sağlayan pencere gösterilmiştir.



Şekil 5.5. Armatür tipi seçimi

Yatay ölçüm armatürün kendi eksenini etrafındaki hareketidir. Düşey ölçüm ise sağdan sola doğru olan hareketidir. Yatay ölçüm sayısı ve düşey ölçüm sayısı alanları yatay ve düşey hareketi sağlayan motorların kaç ölçüm adımında ölçümü tamamlayacaklarını belirler. Şekil 5.6'da yatay ölçüm sayısı ve düşey ölçüm sayısı alanları gösterilmiştir.



Şekil 5.6. Yatay ölçüm sayısı ve düşey ölçüm sayısı alanları

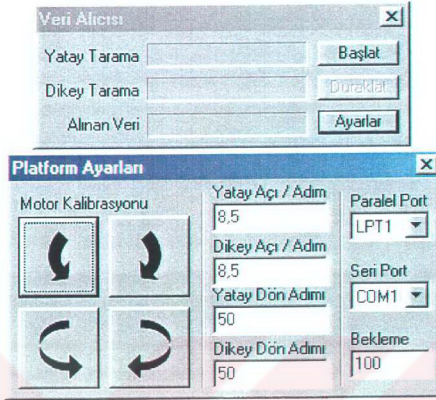
Veri alıcısı alanından yatay tarama ve düşey tarama alanlarına geçiş yapılabilmektedir. Şekil 5.6'da yatay tarama ve düşey tarama alanları gösterilmiştir. Yatay ve düşey hareketi sağlayan motorların programa girilen adım sayısına göre bitirme zamanı karşısındaki boşluğa göre orantısız olarak görülebilmektedir. Bu alınan verilerin takibi ve motorlardan o an hangisinin çalıştığı bu alandan öğrenilmektedir.

Veri alıcısı alanından ayarlar alanına geçiş yapılabilir. Bu alanın motor kalibrasyonu bölümü düşey ve yatay motorun üzerindeki işaretlere göre sağa sola dönmesinin kontrolü için konulmuştur. “↶” işareti düşey eksende hareket eden motoru sağa doğru döndürür. “↷” işareti düşey eksende hareket eden motoru sola doğru döndürür. “↶” işareti yatay eksende hareket eden motoru sağa doğru döndürür. “↷” işareti yatay eksende hareket eden motoru sola doğru döndürür. Veri alıcısı şekil 5.7’de gösterilmiştir



Şekil 5.7. Veri alıcısı

Yatay açı/Adım ve Yatay açı/Adım alanları her adımda motorun döneceği derece değerini belirler. Düşey dönme adımı ve yatay dönme adımı alanları motorların döneceği adım sayısını sınırlandırır. Şekil 5.8’de veri alıcısı ve platform alanları gösterilmiştir.



Şekil 5.8. Veri alıcısı ve platform alanları

Sistem ile ölçüm yapmadan önce sistemin bağlı olduğu bilgisayarın paralel ve seri portunun adresleri belirlenmelidir. Bir bilgisayarın dört seri ve iki paralel port girişi vardır. Bu adreslerin hangisi kullanılıyorsa port adresleri programa bu alandan seçilerek girilmelidir. Seçilmediği takdirde paralel port adresi LPT1 ve seri port adresi COM1 olarak sistem seçer. Şekil 5.9'da Platform ayarları alanı seri ve paralel port seçimi gösterilmiştir. Bekleme alanında her ölçüm için motorun döndüğü açıda bekleme süresi değeri belirlenmektedir.



Şekil 5.9. Platform ayarları alanı ile seri ve paralel port seçimi penceresi

Daha sonra veri alıcısı alanındaki başlat alanına basarak goniofotometre çalıştırılır. Girilen ölçüm sayısına ve armatürün geometrik yapısına bağlı olarak 40 sn ile 240 sn arasında bir bekleme süresi ile sistem bir armatür için gerekli fotometrik ölçüm değerleri olarak ışık şiddet verilerini hesaplayacaktır. Ölçümleri tamamlanan armatürün ışık şiddet verileri, istenilen bir dosya içerisine ölçüm bittiği zamanı gösteren yıl, ay ve gün kısaltmalarıyla .gno uzantılı olarak saklanır.

5.1.2. Veri havuzu konsolu

Bu konsolda fotometrik ölçüm değerlerine bağlı yazılımla hesaplanan ışık şiddet verileri denklem, eğri ve üç boyutlu görüntülere çevrilir. Işık şiddet veri değerleri çevir alanıyla değiştirilebilir. Denklem derecesi veya denklem sayısı bu alandan belirlenebilir. Ayrıca ışık şiddet verilerinin ışın izleme ve ışın etkinlik metotlarıyla bilgisayar ortamında oluşturulan gerçeği taklit eden alanlarda kullanılmasını sağlar. Önceden kaydedilmiş .gno uzantılı dosyalar yeni bir isimle kaydedilmek istenilirse konsol üzerindeki yeni alanı kullanılır. O an ölçümleri tamamlanan armatürün adı bilgisayar tarafından ölçüm bittiği zamanı gösteren yıl, ay, ve gün kısaltmalarıyla atanır. Kullanıcı bu isim yerine dosyaya yeni bir isim vermek istiyorsa yeni ad alanı kullanarak istediği ismi atayabilmektedir. Sil alanı ise .gno uzantılı dosyaları silmek için kullanılır. Şekil 5.10'da veri havuzu konsolu gösterilmiştir.



Şekil 5.10. Veri havuzu konsolu

5.2. Işık Şiddet Eğrilerinin Matematiksel Olarak Tanımlanması

Bir interpolasyon metodu çeşidi olan en küçük kareler metodu, gonifotometrenin yatay ekseninde aldığı ışık şiddet verilerinden eğri çizdirilmesi ve en iyi yaklaşımı sağlayan matematiksel denklemlerin bulunmasında kullanılmıştır. En küçük kareler metodu ile ışık şiddeti vektör büyüklüklerine göre abak üzerine işaretlenen noktaları istenilen dereceden bir polinom fonksiyonu belirlenmektedir.

5.2.1. En küçük kareler metodu

İşaretlenen noktaların oluşturduğu fonksiyonun gerçek değerleri olan ordinatlarla, polinom yaklaşımının değerleri (ordinatları) arasındaki farka sapma denir. En iyi yaklaşımı bulabilmek için sapma miktarları toplamının minimum olması gerekir.

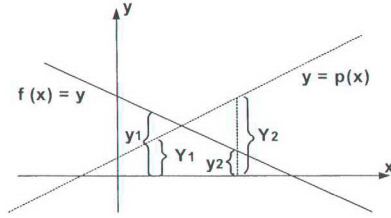
Verilen noktaları (x_i, y_i) ve yaklaşımın noktaları (x_i, Y_i) olsun. Bu takdirde,

$$\sum_{i=1}^n (Y_i - y_i) = \text{minimum} \quad [5.1]$$

şartı yazılır. Eğer $n=2$ alınırsa

$$\sum_{i=1}^2 (Y_i - y_i) = \text{minimum} \quad [5.2]$$

sağlanır. Ancak şekil 5.11’de gösterildiği gibi $y=p(x)$ interpolasyon doğrusu ile $y=f(x)$ esas doğru arasında hiçbir yakınlık yoktur.



Şekil 5.11. İnterpolasyon eğrilerinin gösterilmesi

Bu nedenle Eşitlik 5.1 yerine,

$$\sum_{i=1}^n |Y_i - y_i| = \text{minimum} \quad [5.3]$$

ifadesini almak gerekir (48). Ancak mutlak değerli fonksiyonların minimumdan geçtikleri noktada türevleri olmadığından, minimum bulmak için türev alınmaz. İşte bu nedenle türev alınmasına imkan veren Eş. 5.2 ifadesine denk olan,

$$S = \sum_{i=1}^n (Y_i - y_i)^2 = \text{minimum} \quad [5.4]$$

ifadesi ele alınır. Böylece minimumu bulunan kısmi türevi alınabilen ifade elde edilir.

5.2.2. En küçük kareler metodu ile verilen noktalardan geçen polinom denkleminin bulunması

(x_i, y_i) , $i=1(1)n$, verilen noktalar olmak üzere bu noktalardan geçen ve m ($m \leq n$) dereceden olan bir polinom eğrisini en küçük kareler metoduyla bulunursa;

$$Y = C_0 + C_1x + C_2x^2 + \dots + C_mx^m \quad [5.5]$$

Çizelge 5.1'de. açılara göre ışık şiddet verilerinin ışık şiddet eğrileri Eş. 5.4'deki gibi ifade edilmektedir. Denklemdaki C katsayılarının belirlenmesi için her nokta denklemini minimize yapan denklemlerin kısmi türevi alınarak sıfıra eşitlenirse;

Çizelge 5.1. Açılara göre ışık şiddet verilerinin gösterilmesi

Veri Modellemesi - 2003.06.17 11:37											
Yatay Veri	1	2	3	4	5	6	7	8	9	10	11
1	479	946	1749	2126	2317	2009	3003	3692	4076	4571	4892
2	766	1383	1754	1776	2342	2789	2959	3396	4148	4369	4696
3	340	860	1609	2091	2594	2701	3291	3573	3840	4667	5126
4	458	970	1319	1969	2444	2743	3438	3398	3695	4568	4787
5	421	948	1437	1874	2432	2960	3421	3407	4149	4685	4951
6	471	1021	1700	1824	2579	2959	3257	3383	4174	4490	4836
7	744	1268	1723	1992	2522	2696	3254	3447	3901	4427	5051
8	525	1211	1345	2063	2400	2754	3391	3640	4095	4532	4670
9	500	1158	1750	2140	2568	2682	3186	3829	4103	4554	4950
10	969	1264	1548	1637	2281	2570	3250	3630	3898	4529	4989
11	430	1211	1661	1630	2591	2904	3345	3458	4232	4379	4937
12	706	1268	1530	1726	2397	2906	3301	3799	3834	4422	5030
13	903	1114	1280	1797	2260	2077	3077	3442	4227	4567	4929
14	941	1219	1371	2067	2501	2591	3690	3673	3996	4347	5095
15	706	1271	1477	2052	2139	2663	3384	3817	4146	4471	4851
16	689	1161	1684	2091	2534	2577	3093	3591	4106	4696	4893
17	583	1273	1669	2128	2276	2947	2970	3623	4261	4379	4717
18	952	955	1729	1792	2211	2947	3276	3813	4089	4422	4953
19	763	967	1380	1879	2422	2590	3171	3500	3890	4694	4895

$$S = S(C_0, C_1, \dots, C_m) = \sum_{i=1}^n (Y_i - y_i)^2 \quad [5.6]$$

olacaktır. Buradan;

$$\frac{\partial S}{\partial C_0} = \sum_{i=1}^n 2(C_0 + C_1 x + \dots C_m x^m - y_i) = 0$$

$$\frac{\partial S}{\partial C_1} = \sum_{i=1}^n 2(C_0 + C_1 x + \dots C_m x^m - y_i) x_i = 0 \quad [5.7]$$

Λ

$$\frac{\partial S}{\partial C_m} = \sum_{i=1}^n 2(C_0 + C_1 x + \dots C_m x^m - y_i) x_i^m = 0$$

sistemi bulunur. Bu sistem daha açık olarak;

$$nC_0 + \left(\sum_{i=1}^n x_i\right)C_1 + \left(\sum_{i=1}^n x_i^2\right)C_2 + \dots + \left(\sum_{i=1}^n x_i^m\right)C_m = \sum_{i=1}^n y_i$$

$$\left(\sum_{i=1}^n x_i\right)C_0 + \left(\sum_{i=1}^n x_i^2\right)C_1 + \left(\sum_{i=1}^n x_i^3\right)C_2 + \dots + \left(\sum_{i=1}^n x_i^{m+1}\right)C_m = \sum_{i=1}^n x_i y_i$$

$$\left(\sum_{i=1}^n x_i^2\right)C_0 + \left(\sum_{i=1}^n x_i^3\right)C_1 + \left(\sum_{i=1}^n x_i^4\right)C_2 + \dots + \left(\sum_{i=1}^n x_i^{m+2}\right)C_m = \sum_{i=1}^n x_i^2 y_i$$

$$\dots \dots \dots \dots \dots \dots$$

$$\left(\sum_{i=1}^n x_i^m\right)C_0 + \left(\sum_{i=1}^n x_i^{m+1}\right)C_1 + \left(\sum_{i=1}^n x_i^{m+2}\right)C_2 + \dots + \left(\sum_{i=1}^n x_i^{m+m}\right)C_m = \sum_{i=1}^n x_i^m y_i$$
[5.8]

şeklinde yazılır. Bu sistem $(m+1)$ tane denklem ve $(m+1)$ tane bilinmeyenden oluşmuştur ve bu denklemler oluşturulan yazılım ile bilgisayar tarafından çözülür (49). Böylece elde edilen $C_0, C_1, \dots, C_m$ değerleri belirlenir. Yazılımla hesaplatılan C katsayıları Çizelge 5.2'de gösterilmiştir.

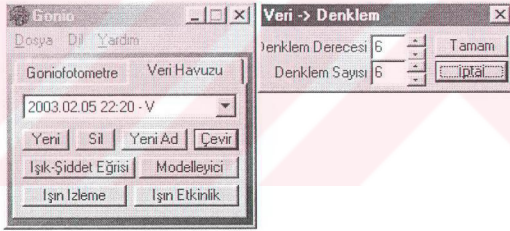
Çizelge 5.2 Hesaplanan C katsayılarının gösterilmesi

ışık Akısı	1000	Ölçüm Uzaklığı	1	Lamba Türü	Akkor	☒	
Denklem	X^9	X^8	X^7	X^6	X^5	X^4	X^3
1	-3.48	0,002	0,49	-64,18	2162,15	68958,96	522184,34
2	-3.48	0,002	0,61	-75,37	2649,71	63545,76	448794,46
3	-3.48	0,001	0,73	-88,57	3310,89	50291,18	515713,40
4	-3.55	0,002	0,68	-85,46	3330,39	42525,00	693115,12
5	-3.32	0,002	0,46	-62,76	2477,43	50429,18	786789,93
6	-2.82	0,002	0,22	-35,46	1313,55	69755,70	705559,75
7	-2.59	0,002	0,18	-28,32	876,52	83184,60	549414,56
8	-3.19	0,002	0,43	-55,46	1833,23	74649,03	512389,59
9	-4.04	0,002	0,71	-88,42	3119,46	56154,35	606234,18

5.2.3. Modelleyici alanı

Şekil 5.10'daki veri havuzu konsolundan modelleyici alanına ulaşılabilir. Modelleyici alanından, Çizelge 5.1'deki açılara göre ışık şiddet verilerinin gösterilmesi ve Çizelge 5.2' deki gösterilen ışık şiddet eğri denklemlerinin C katsayıları denklemlerine bağlı olarak kullanıcı isteğine göre hesaplanır.

C katsayıları gonio konsolundan girilen yatay ve düşey eksen adım sayıları kadar veriyi tabloda gösterir. Ayrıca program oluşacak olan üç boyutlu görüntünün kaç denklemlerle tanımlanacağı ve kaçınıcı dereceden denklemler ile birleştirileceğini kullanıcının isteğine göre belirleyebilir. Bunun için şekil 5.12'deki denklemler ve denklemler derecesi belirleme alanına yazarak alan üzerindeki tüm butonlara basılır. Veri havuzu konsoluna döndürülür, modelleyici butonuna basılarak belirlenen denklemler sayısında ve derecesindeki C katsayılarına ulaşılabilir.



Şekil 5.12. Denklem sayısı ve derecesi belirleme

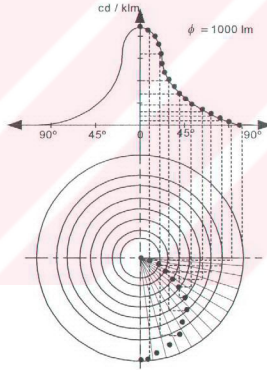
5.3. Işık Şiddeti Verilerinden Eğri ve Üç Boyutlu Görüntü Oluşturma

Işık şiddeti eğrileri kullanılarak görüntü oluşturulması işlemine görüntü sentezi denir (45). Bilgisayar grafiği, aydınlatma konusunu yerel, genel ve dolaylı aydınlatma olarak üçe ayırır. Belirli kabullere ve yazılım dili seviyesine göre ışık kaynağı modelleri oluşturulmaktadır. Görüntü sentezi yapılırken genel aydınlatmada kullanılan metod kernel şiddet tahminidir (46). Oluşturulan bilgisayar ortamlarında kurulan sistemdeki goniofotometre ve yazılım gerçek armatürleri tanımlayan

matematiksel modellerin belirlenmesinde kullanılır. Böylece aydınlatma tasarım programlarıyla armatür seçimi ve tasarımı kolaylaştırılmıştır (47).

5.3.1. Işık şiddet eğrisinin elde edilmesi

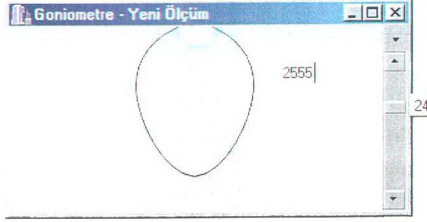
Yazılımda ışık şiddet verileri, 1000 lümenlik ışık akısına göre sabitlenerek hesaplanmıştır. Böylece eğrilerin ve üç boyutlu ışık dağılım yüzeyinin üzerine çizdirildiği abak, tüm armatürlerin karşılaştırılabildiği bir alan haline gelmiştir. İç içe geçen on halkadan oluşan abakta her bölge belli bir değer aralığındadır. Kullanıcı böylece bir açıya bağlı ışık şiddet değerini cd/klm olarak abak üzerinden belirleyebilir. Şekil 5.13’de ışık şiddet vektörlerinin abak üzerine aktarılması gösterilmiştir.



Şekil 5.13. Işık şiddet vektörlerinin abak üzerine aktarılması

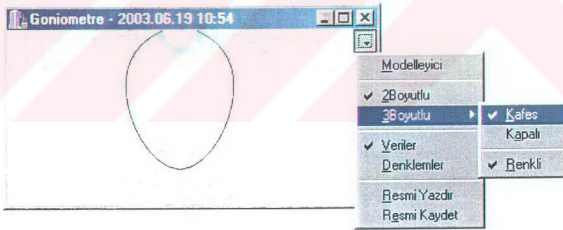
Şekil 5.23’ deki gibi eğrinin sağ tarafında bulunan dikdörtgen sembol, fare ile yukarı aşağı hareket ettirilirse 0 ile 90 derece arasında bir armatüre ait tüm eğriler görülebilir. Doğrusal armatürlerde 90 ile 180 derece arasında geri dönüşüm eğrileri de bu eğrilerin aynısı olacaktır. Benzer olarak yüzeysel armatürün tüm açılardaki eğrileri de incelenebilir.

Bu eğri üzerinde mouse hareket ettirilirse mouse'un bulunduğu her noktada açığı bağılı cd / klm değeri sarı bir kutu içinde ekrandan şekil 5.14'deki gibi görülebilir.



Şekil 5.14. İki boyutlu ışık şiddet eğrisi

Bir armatüre ait oluşturulan üç boyutlu görüntülere geçiş için şekil 5.15'deki eğrinin sağ üst köşesinde bulunan ok ikonunu tıklanarak sağlanabilir. Bu ulaşım noktasından denklemlere, verilere geçiş de sağlanabilir. Ayrıca ışık şiddet eğrisi bilgisayardaki bir dosyaya resmi yazdır ve resmi kaydet ile yazdırılabilir ve kaydedilebilir.



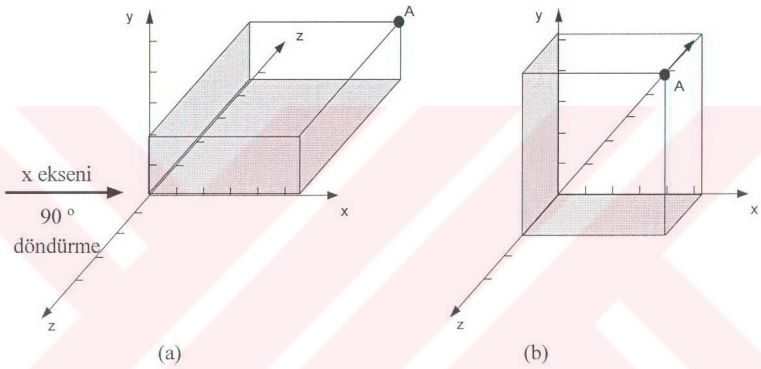
Şekil 5.15. İki boyutlu ışık şiddet eğrilerinden üç boyutlu hacimlere geçiş

5.3.2. En küçük kareler metodu yöntemiyle elde edilen eğrilerden üç boyutlu görüntü oluşturma

Bir armatüre ait goniofotometre ile elde edilen ışık şiddet verilerinden yazılımdaki dönüşüm formülleriyle üç boyutlu ışık şiddet görüntüsü oluşturulmaktadır.

5.3.2.1. Bir noktayı üç boyutta döndürme

Üç boyutlu koordinat sisteminde tanımlanmış olan bir noktanın belirli bir döndürmeden sonraki yeni koordinatları dönüşüm matrisleri ile bulunabilir (50). Şekil 5.16'da a ve b'de bir A noktasının x eksenine etrafındaki 90° derece döndürülmesi ile belirlenen koordinatları gösterilmektedir.



Şekil 5.16. (a)'da A noktası 0° derecede ve (b)'de bir A noktasının x eksenine etrafındaki 90° derece döndürülmesi ile belirlenen koordinatları

Orijinden bakıldığında saat dönüş yönü pozitif yön olarak kabul edilirse, X eksenine etrafında bir θ açısı için dönüşüm matrisi;

$$[T_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [5.9]$$

Y eksenine etrafında bir β açısı için dönüşüm matrisi

$$[Ty] = \begin{bmatrix} \cos\beta & \sin\beta & 0 & 0 \\ -\sin\beta & \cos\beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [5.10]$$

Z eksenini etrafında bir φ açısı için dönüşüm matrisi

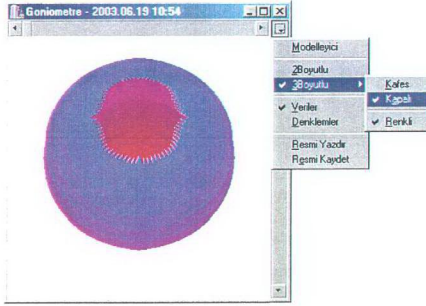
$$[Tz] = \begin{bmatrix} \cos\varphi & 0 & -\sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [5.11]$$

[5.9], [5.10] ve [5.11]. Eşitliklerden dönüşüm matrisleriyle bir noktanın üç boyuttaki tüm açılarına göre değeri matrislerin yazılımdaki bilgisayar çözümlerinden belirlenmektedir.

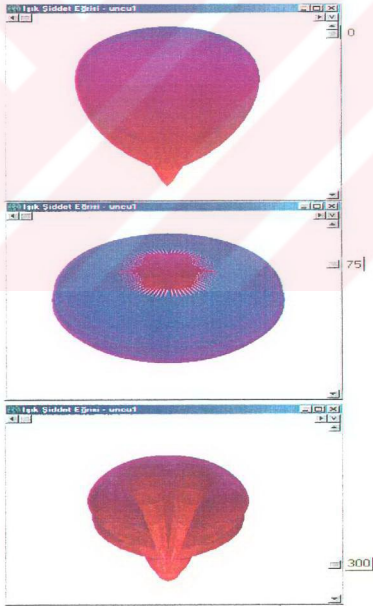
5.3.2.2. Üç boyutlu ışık dağılım yüzeyi

Bir armatürün düşey ekseninde döndürülerek yatay eksen verilerinin iki boyutlu ışık şiddet eğrilerinin oluşturmuş olduğu üç boyutlu hacim şeklindeki gibi kapalı ve renkli bir şekilde görsel hale getirilebilir. Bu ışık dağılım yüzeyi üzerinde fare hareket ettirilirse mouse'un bulunduğu her noktada açıya bağlı cd/klm değeri sarı bir kutu içinde ekrandan kullanıcı tarafından görülebilir.

Şekil 5.17'de üç boyutlu ışık dağılım yüzeyi gösterilmiştir. üç boyutlu ışık dağılım yüzeyinin sağ tarafındaki ikonun mouse ile hareket ettirilmesi ile hacmi 360^0 derece kendi etrafında döndürerek hacim yapısını inceleme imkanı verir. Şekil 5.18'de üç boyutlu ışık dağılım yüzeyinin kendi etrafında döndürülerek incelenmesi gösterilmiştir. Bu hacim yapısı kırmızı ile mavi arasında kurulmuş olup ışık şiddeti az olan noktalarda mavinin tonları, ışık şiddeti çok olan noktalarda kırmızının tonları atanarak ışık şiddeti verileri görsel hale getirilmiştir.



Şekil 5.17. Üç boyutlu ışık dağılım yüzeyi

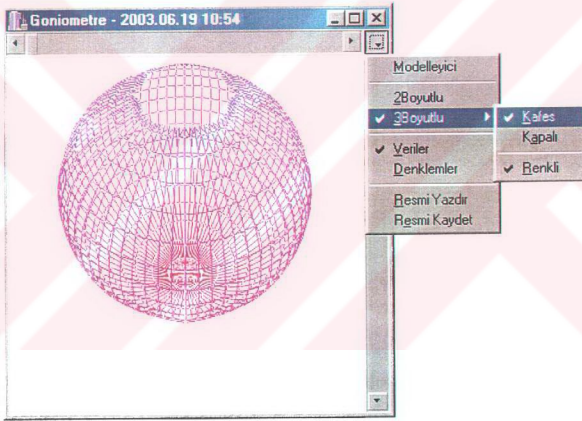


Şekil 5.18. Üç boyutlu ışık dağılım yüzeyinin kendi etrafında döndürülerek incelenmesi

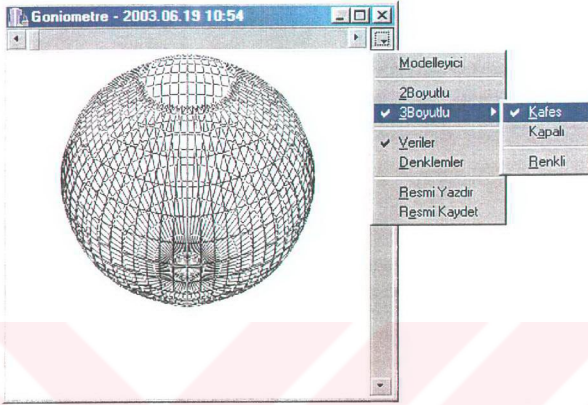
Şekil 5.19'daki gibi sağ üst butonları kullanılarak üç boyutlu ışık dağılım yüzeyi renkli kafes biçimli olarak görsel hale getirilebilir.

Şekil 5.20'daki gibi sağ üst butonları kullanarak üç boyutlu ışık dağılım yüzeyi siyah kafes biçimli olarak görsel hale getirilebilir.

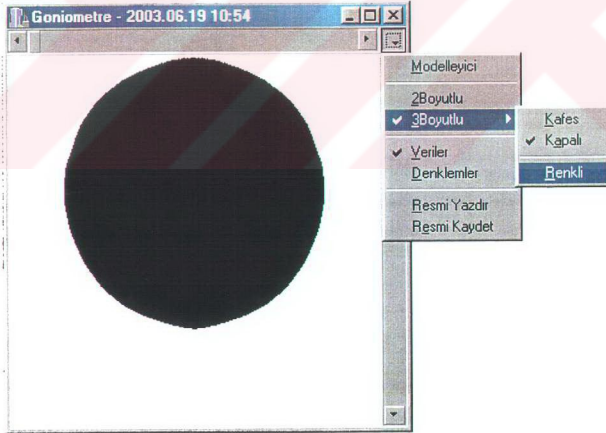
Sağ üst butonları kullanılarak renkli seçeneği kaldırılırsa Şekil 5.21'de üç boyutlu ışık dağılım yüzeyi katı biçimli gösterildiği gibi hacmi inceleme alanları oluşturulmaktadır.



Şekil 5.19. Üç boyutlu ışık dağılım yüzeyinin renkli kafes biçiminde gösterilmesi



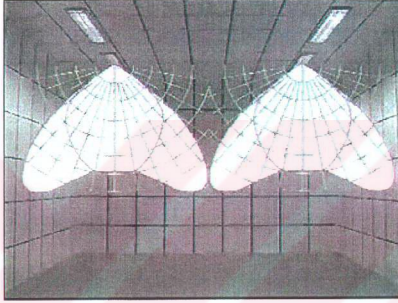
Şekil 5.20. Üç boyutlu ışık dağılım yüzeyinin siyah renkli kafes biçiminde gösterilmesi



Şekil 5.21. Üç boyutlu ışık dağılım yüzeyinin katı biçimli gösterilmesi

5.3.3. Üç boyutlu uygulama alanları

Üç boyutlu uygulama alanlarında şekil 5.22'deki gibi ışık şiddet eğrilerinin gerçek oda içerisine dağılımında gösterildiği gibi ışık şiddeti verilerinin oda içerisine dağılımı amaçlanmıştır.



Şekil 5.22. Işık şiddet eğrilerinin gerçek oda içerisine dağılımının gösterilmesi

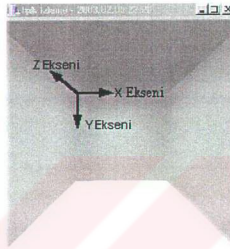
Gonio konsolu üzerindeki ışın izleme alanı ve ışın etkinlik alanı üç boyutlu uygulama alanlarına geçiş için kullanılır. Şekil 5.23'de gonio konsolundan ışın izleme ve ışın etkinlik alanlarına geçişin yapılacağı alanlar gösterilmiştir.



Şekil 5.23. Gonio konsolundan ışın izleme ve ışın etkinlik alanlarına geçiş

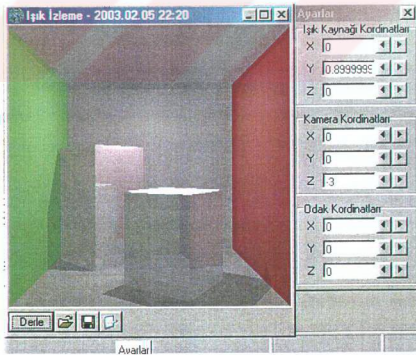
5.3.3.1. Ayarlar penceresi

Kullanıcı ayarlar penceresinden  ışık kaynağını, kamera ve odak ayarı yapabilir. Şekil 5.24' de üç boyutlu ortamda X, Y ve Z eksenlerini gösterilmiştir.



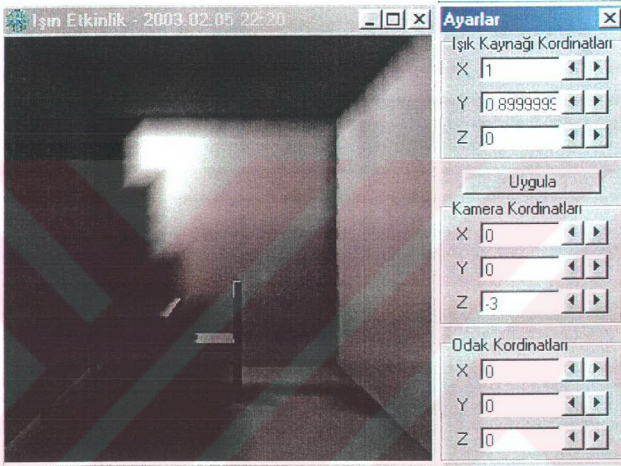
Şekil 5.24. Üç boyutlu ortamda X, Y ve Z eksenlerinin gösterilmesi

Üç boyutlu ortam içerisinde derle, dosyadan yükle, resmi kaydet ve ayarlar penceresi vardır. Şekil 5.25' de örnek alan ve ayarlar penceresi gösterilmiştir.



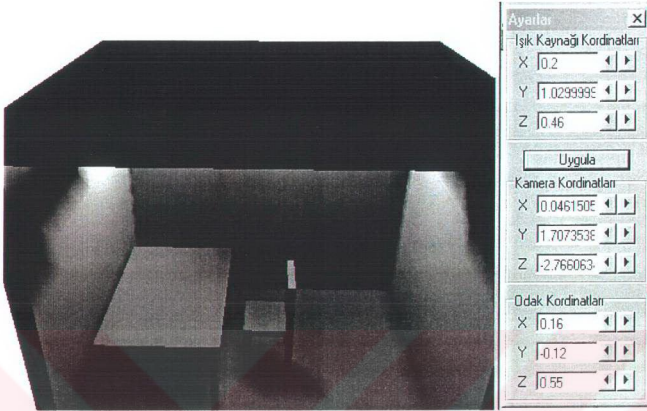
Şekil 5.25. Örnek alan ve ayarlar penceresi

Ayarlar penceresi, ışık kaynağı koordinatlarından ışık kaynağının X, Y, ve Z düzlemlerindeki yeri belirlenebilmektedir. Böylece örnek odanın herhangi bir yerine ışık kaynağı yerleştirilerek örnek odanın görüntüsü izlenebilir. Şekil 5.26'da ışık kaynağı koordinatlarındaki değişikliklerin örnek alan üzerine etkileri gösterilmiştir.



Şekil 5.26. Işık kaynağı koordinatlarındaki değişikliklerin örnek oda üzerine etkileri

Ayarlar penceresindeki kamera koordinatları X, Y, ve Z düzlemlerindeki yerini belirleyebilmektedir. Böylece kullanıcı örnek odaya istediği yönden bakabilir. Şekil 5.27'de kamera koordinatlarındaki değişikliklerin örnek alana etkisi gösterilmiştir.



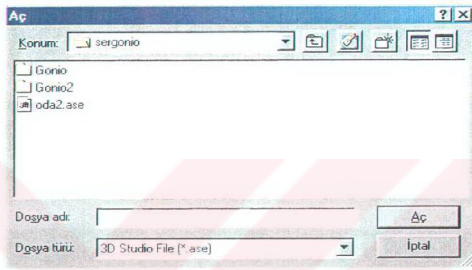
Şekil 5.27. Kamera koordinatlarındaki değişikliklerin örnek alana etkisi

Ayarlar penceresindeki odak koordinatları kullanıcının ufuk çizgisini belirler. X, Y, ve Z düzlemlerindeki yerini belirleyerek örnek alanı sağa ve sola kaydırıp imkanı sağlar. Z eksenindeki hareket ettirerek yakınlaştırma ve uzaklaklaştırma özelliği kazandırır. Kullanıcı görüntüden bağımsız olarak çalışan ışın etkinlik metotunda ayarlar penceresini kullanabilmektedir. Işın izleme metotunda ayarlar penceresi kullanılamamaktadır.

5.3.3.2. Üç boyutlu ortam yükleme

Örnek alan kutusu dışında sisteme autodesk yazılım grubunda üretilen autocad ve 3D studio max paket programlarında üretilen bir birim bir metre olarak ölçeklendirilen sanal ortamlar yüklenebilir. Işık şiddet eğrilerini temel alan sekiz paket programdan yazılımın farkı, mekanlar ve ışık kaynakları sanal olmasına karşın kurulan sistemin goniofotometre bölümüyle elde edilen armatür verilerinin sanal ortamlarda denenerek uygulama yapılmadan armatür ve mekan hakkında bir fikir edinilebilmesidir.

Kullanıcı 3D studio max paket programı ortamında .ase uzantılı olarak dosyaya kaydederek yapmış olduğu örnek alanı sistem içine alma imkanı bulur. ASE uzantısı ASCII Scene Export kelimelerinin baş harfleridir. Şekil 5.28'de dosyadan üç boyutlu ortam yükleme işlemi gösterilmiştir.

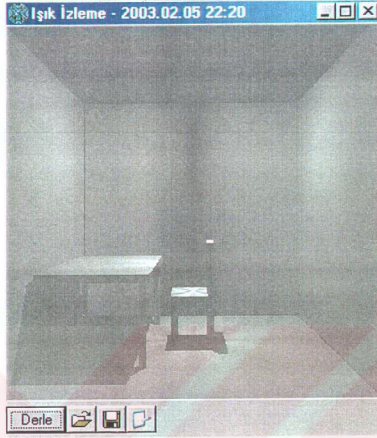


Şekil 5.28. Dosyadan üç boyutlu ortam yükleme

5.3.3.2.1 Işın izleme alanı

Gonio konsolundan ışın etkinlik alanına geçiş yapılırken ışın izleme alanı için örnek alan sisteme direk olarak yüklenir. Örnek alanın karşılıklı duvarlarının farklı renklerde olması ve oda içerisindeki nesnelerin birbirine bu renkleri armatürün ışık şiddet vektörleri değeriyle oda içerisindeki eşyaların görünümünü etkilediği göstermektedir.

Gerçek bir ortama giren ışığın sonsuz kere yansıdığı kabul edilir. Derleme sayısı artırılınca oda içerisine dağılan ışığın gerçeğe yakınlığı artacaktır. Şekil 5.29'da 3d studio max programıyla oluşturulan örnek alana armatür ışık şiddeti verilerinin ışın izleme yöntemi görüntü oluşturulması gösterilmiştir.



Şekil 5.29. 3d studio max programıyla oluşturulan örnek alanda ışın izleme yöntemi görüntü oluşturulması

5.3.3.2.2 Işın etkinlik alanı

Genel ışın etkinlik metodunun formüle edilmesi ve çözümü için gerekli temel aşamalar aşağıdaki gibidir:

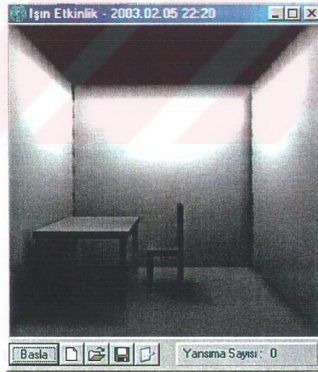
1. Alanlar alt elemanlara bölünür.
2. Elemanlar üzerinde kesin ışın etkinlik değerlerinin hesaplanacağı bölgeler veya düğümler seçilir.
3. Bir eleman üzerinde değişken ışın etkinlik değerleri için basit fonksiyonel bir form bulunur.
4. Sonlu bir hata ölçütü seçilir. Bu hatayı sonlu bir toplama yaklaştırır. Böylece sonsuz boyutta olan problem sonlu sayıda lineer eşitliğe dönüşür.
5. Lineer sistemin katsayıları hesaplanır. Form faktörü olarak adlandırılan bu katsayılar, elemanlar arası ışığın geçmesini sağlayan geometrik ilişkilere bağlıdır.

6. Bilinmeyen noktasal ışın etkinlik değerler için lineer eşitlikler çözülür. Yaklaşık ışın etkinlik sonucu, noktasal ışın etkinlik değerleri ile ağırlıklı oluşturulan taban fonksiyonların lineer toplamı şeklinde yeniden oluşturulur.

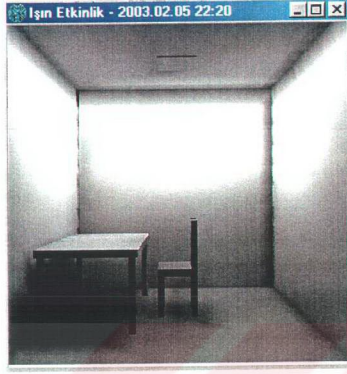
7. İstenilen bakış yönü için ışın etkinlik yaklaşımından elde edilen gölgeleme değerleri kullanılarak görüntü oluşturulur.

Gonio konsolundan ışın etkinlik alanına geçiş yapılnca örnek alan sisteme direk olarak yüklenir. Kullanıcı isterse 3d max studio ortamda oluşturulan odayı örnek alana yükleyebilir. Goniofotometre kısmıyla elde edilen armatürlerin ışık şiddetverileri dosyadan ışın etkinlik yöntemi ile görüntü oluşturulur.

Aşağıdaki şekillerde ışın etkinlik yöntemiyle bir armatürün verilerinin üç boyutlu sanal bir alana yüklenmesinin şekil 5.30'da başlangıç konumu, şekil 5.31'de ışın etkinlikte 1607 yansımadan sonra oluşan ışık dağılımı gösterilmiştir.



Şekil 5.30. Işın etkinlik alanı başlangıç konumu

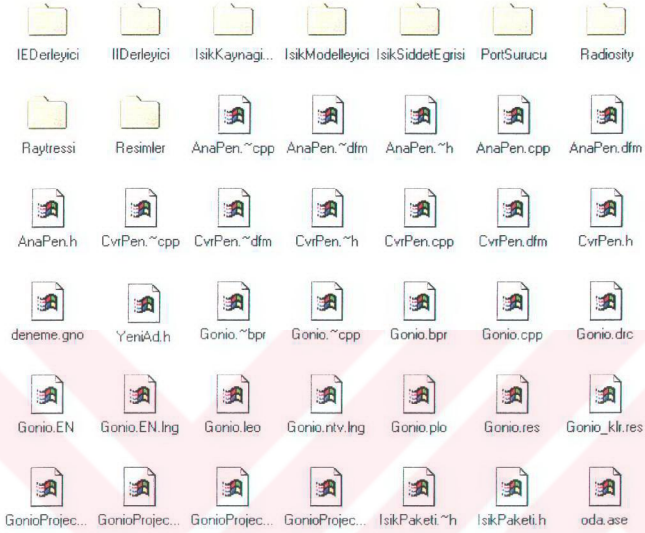


Şekil 5.31. Işın etkinlikte 1607 yansımadan sonra oluşan ışık dağılımı

5.4. Yazılımın Kaynak Kodları

Borland C++ Builder formu diğer bazı programlardan farklı olarak araçlardan, menülerden ve başlık çubuğundan oluşur. Visual alanları yöneten menüler ve formlara geçiş yaptıracak olan başlık çubukları ana form üzerinde bulunur (43).

Bir proje alt programlardan oluşmaktadır. Şekil 5.32’de projeyi oluşturan alt programlar klasörleri gösterilmiştir. Bu gruplar içerisindeki .cpp ve .h uzantılı dosyalar her bir alt alanın kaynak kodlarını içeren dosyalardır(44). Yazılımın kaynak kodları Ek 1’de verilmiştir.



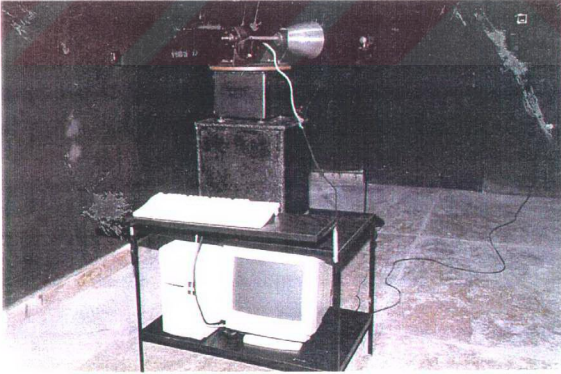
Şekil 5.32. Projei oluşturan alt program klasörleri

6. DENEYSEL SONUÇLAR

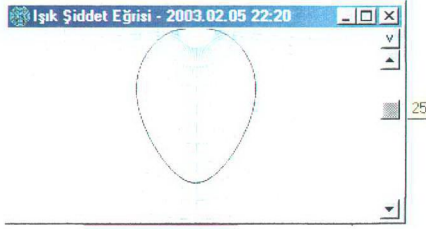
Şekil 6.1'de bir ışık şiddeti eğrisi ile temsil edilebilen armatürün fotoğrafı gösterilmiştir. Şekil 6.2'de bir ışık şiddeti eğrisi ile temsil edilebilen armatürün goniofotometrik ölçümleri yapılan sistemin fotoğrafı gösterilmiştir. Sistemle elde edilen eğriler Şekil 6.3'de bir ışık şiddeti eğrisi ile temsil edilebilen armatürün 25° derecedeki ışık şiddet eğrisi ve Şekil 6.4'de bir ışık şiddeti eğrisi ile temsil edilebilen armatürün 79° derecedeki ışık şiddet eğrisi gösterilmiştir. Ayrıca yazılımla elde edilen ışık dağılım yüzeyi Şekil 6.5'de bir ışık şiddeti eğrisi ile temsil edilebilen armatürün 0° derecedeki ışık dağılım yüzeyi gösterilmiştir.



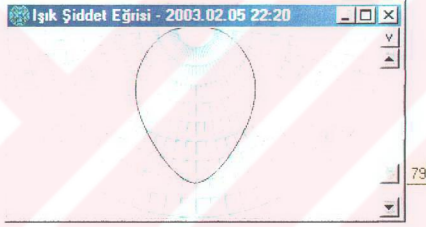
Şekil 6.1. Bir ışık şiddeti eğrisi ile temsil edilebilen armatür



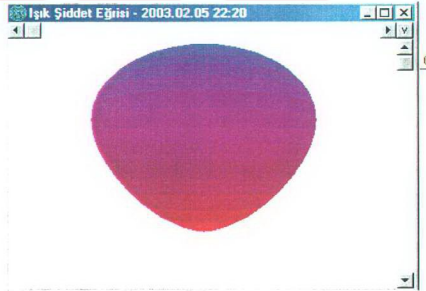
Şekil 6.2. Armatürün goniofotometrik ölçümleri yapan sistem



Şekil 6.3. Bir ışık şiddeti eğrisi ile temsil edilebilen armatürün 25° derecedeki ışık şiddet eğrisi



Şekil 6.4. Bir ışık şiddeti eğrisi ile temsil edilebilen armatürün 79° derecedeki ışık şiddet eğrisi

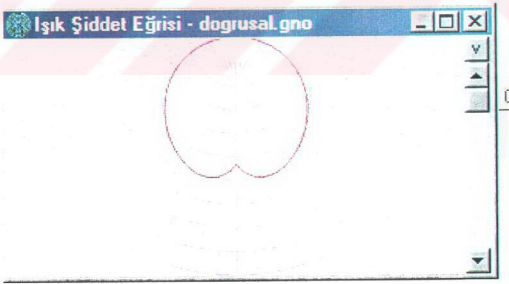


Şekil 6.5. Bir ışık şiddeti eğrisi ile temsil edilebilen armatürün 0° derecedeki ışık dağılım yüzeyi

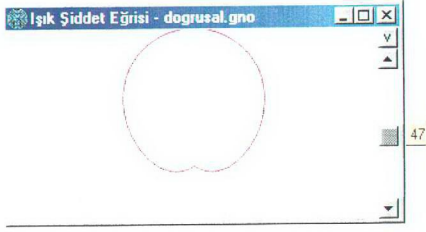
Şekil 6.6'de iki ışık şiddeti eğrisi ile temsil edilebilen armatürün goniofotometrik ölçümlerinin yapıldığı sistemin fotoğrafı gösterilmiştir. Sistemle çıkartılan eğriler şekil 6.7'de iki ışık şiddeti eğrisi ile temsil edilebilen armatürün 0° derecedeki ışık şiddet eğrisi şekil 6.8' da iki ışık şiddeti eğrisi ile temsil edilebilen armatürün 47° derecedeki ışık şiddet eğrisi şekil 6.9'da iki ışık şiddeti eğrisi ile temsil edilebilen armatürün 90° derecedeki ışık şiddet eğrileri gösterilerek iki ışık şiddeti eğrisi ile temsil edilebilen armatür olduğundan eğriler açılara göre farklı çıkmıştır. Ayrıca yazılımla şekil 6.10 ve 6.11'de iki ışık şiddeti eğrisi ile temsil edilebilen armatürün 56° ve 93° derecelerindeki ışık dağılım yüzeyi gösterilmiştir.



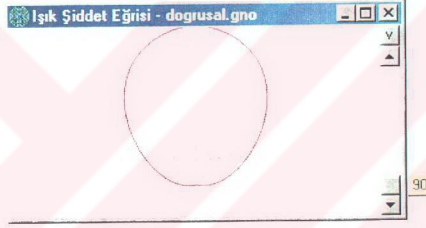
Şekil 6.6. İki ışık şiddeti eğrisi ile temsil edilebilen armatür



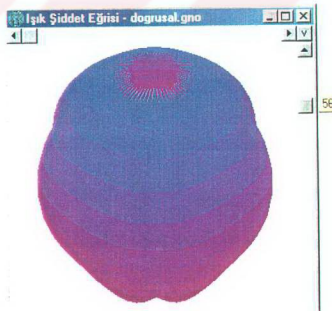
Şekil 6.7. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün 0° derecedeki ışık şiddet eğrisi



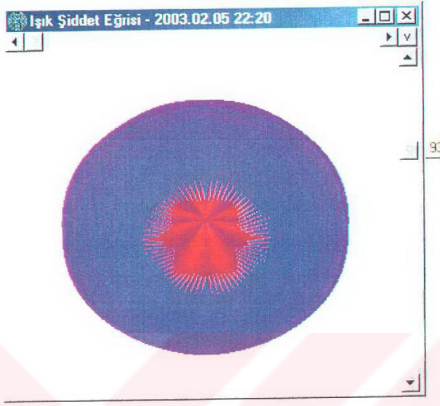
Şekil 6.8. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün 47° derecedeki ışık şiddet eğrisi



Şekil 6.9. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün 90° derecedeki ışık şiddet eğrisi



Şekil 6.10. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün 56° derecedeki ışık dağılım yüzeyi

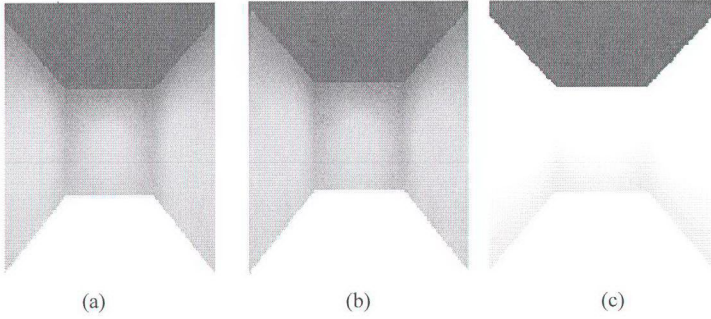


Şekil 6.11. İki ışık şiddeti eğrisi ile temsil edilebilen armatürün 93° derecedeki ışık dağılım yüzeyi

6.1. Işın İzleme Yöntemiyle Duvar Renklerinin Karşılaştırılması

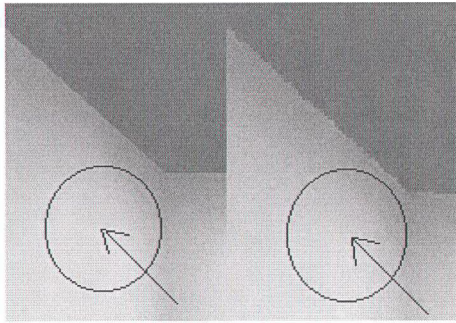
Şekil 6.12' de (a) ve (b)'de aynı lamba farklı reflektör, (c)'de ise (a)'daki reflektörle aynı güçlü lamba benzetimi olmak üzere üç değişik armatürün ışık şiddet verilerini kullanarak yapılan ışın izleme karşılaştırılması örnek bir oda içerisinde aktarılması gösterilmiştir.

Bu odalar tasarlanırken oda içerisindeki ışık şiddetine bağlı aydınlık düzeyi beyaz ile siyah arasında renk aralığı ile sembolize edilmektedir. Beyaz en aydınlık düzeyi yüksek bölgeleri, siyah ise en karanlık bölgeleri oluşturmaktadır.



Şekil 6.12. (a) ve (b)'de aynı lamba farklı reflektör, (c)'de ise (a)'daki reflektörle aynı güçlü lamba benzetimi olmak üzere üç değişik armatürün ışık şiddet verilerini kullanarak yapılan ışın izleme karşılaştırılması

Şekil 6.12 (c)'deki odada ise yüksek ışık akısına sahip bir armatürün ışık şiddet verilerinin ışın izleme yöntemi ile oda içerisine aktarılması gösterilmiştir. Şekil 6.13' da iki değişik armatürün ışık şiddet verilerinden elde edilen ışın izleme aralarındaki farkı gösterir. Şekiller tavana yakın bölgelerin daha açık renkli olduğu büyütülerek gösterilmiştir. Oda içerisindeki beyaz bölgelerin çok olması duvarlar üzerindeki aydınlık düzeylerinin çok yüksek olduğunu gösterir.



Şekil 6.13. Karşılaştırma için tavana yakın kısımlar büyütülmüştür.

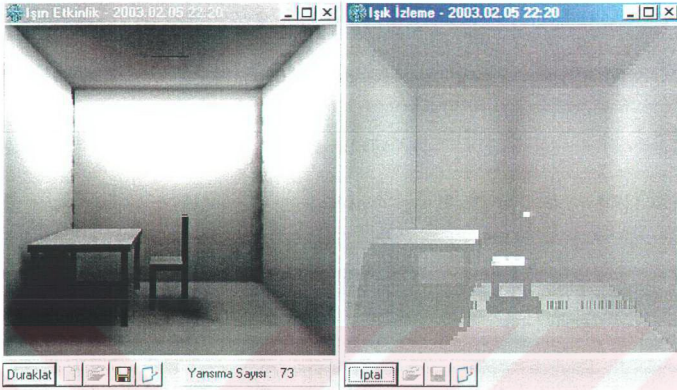
6.2. Işın İzleme ve Işın Etkinlik Metotlarının Karşılaştırılması

Işın izleme ve ışın etkinlik metodlarının algoritmalarının farklı çalışmasından kaynaklanan görüntü farklılıklarının oluşması yazılım içerisinde iki ayrı görüntü aktarma yöntemlerinin kullanılmasını zorunlu kılmıştır. Her iki yöntemin de avantaj ve dezavantajları vardır.

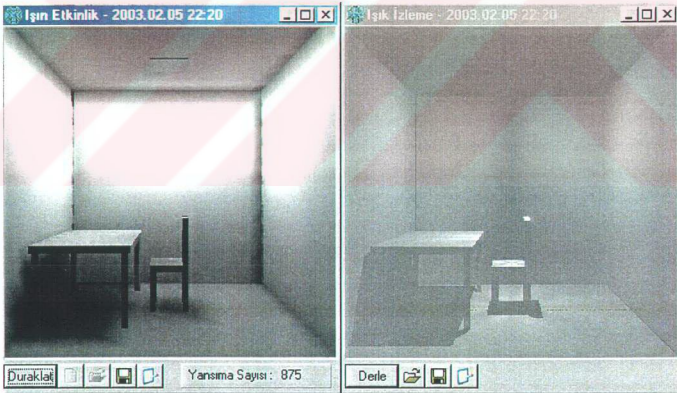
Karşılaştırma sonuçlarına göre ışın izleme yönteminin özellikleri şöyle sıralanabilir; Güçlü ve bir ışık şiddeti eğrisi ile temsil edilebilen ışık kaynakları için kullanılması daha uygundur. Aynasal yansıma gösterdiğinden düzgün ve parlak yüzeyler için tercih edilmelidir. Sadece göze gelen ışınları takip edilir bu yüzden az sayıda ışın takibi gölgeleri olduğundan daha keskin ve koyu olarak tanımlar.

Karşılaştırma sonuçlarına göre ışın etkinlik yönteminin özellikleri de şöyle sıralanabilir; Tüm ışık kaynaklarını modelleyebilir. Fakat ışın izlemeye göre yayınık dağılımlı ışık kaynakları olan iki ışık şiddeti eğrisi ile temsil edilebilen ve üç ve daha fazla ışık şiddeti eğrisi ile temsil edilebilen ışık kaynaklarının benzetimi için daha uygundur. Yayınık sistemlerin alt alanlarını topladığından renk karışımlarını ve renk solmalarını gösterebilir. Yayıma üzerine kurulu algoritması yumuşak gölgeleri gösterebilir. Pürüzlü alanların yansımalarını modelleyebilir.

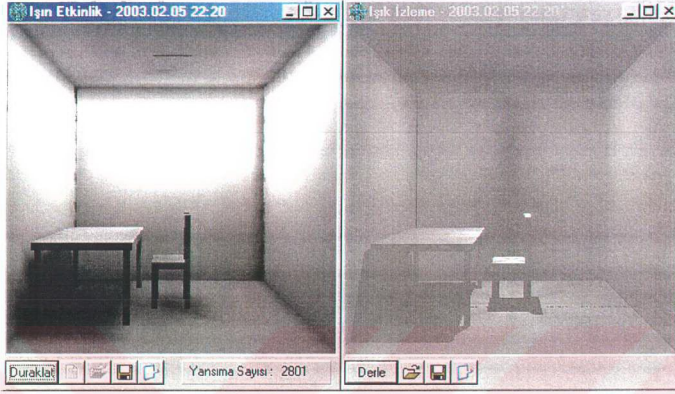
Şekil 6.14, şekil 6.15, şekil 6.16 ve şekil 6.17’de ışın izleme ve ışın etkinlik metodları 73, 875, 2801, 3132 yansımadan sonra oluşan ışık dağılımı gösterilmiştir.



Şekil 6.14. Işın izleme ve ışın etkinlik metodlarının 73 yansımadan sonra oluşan ışık dağılımı



Şekil 6.15. Işın izleme ve ışın etkinlik metodlarının 875 yansımadan sonra oluşan ışık dağılımı

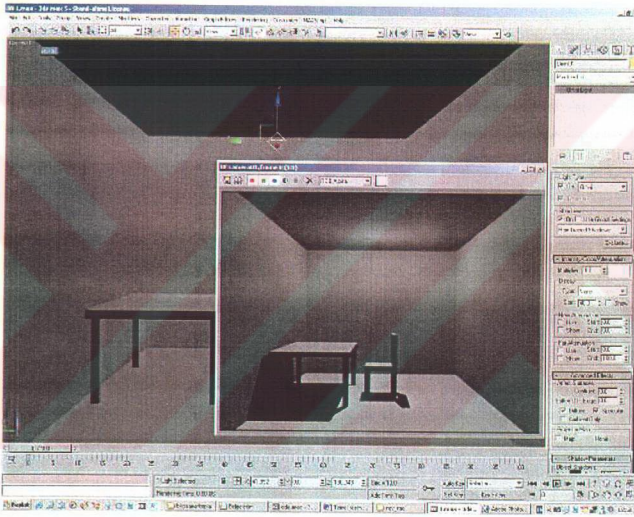


Şekil 6.16. Işın izleme ve ışın etkinlik metodlarının 2801 yansımadan sonra oluşan ışık dağılımı



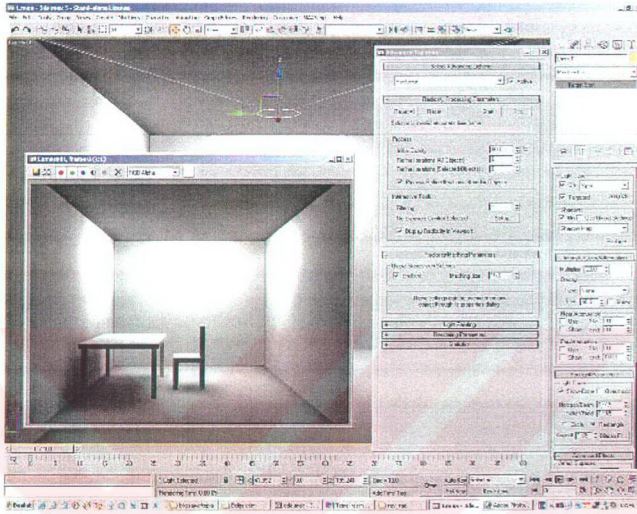
Şekil 6.17. Işın izleme ve ışın etkinlik metodlarının 3132 yansımadan sonra oluşan ışık dağılımı

Yazılım ile elde edilen görüntülerin doğruluğu diğer programlarla karşılaştırma yapılarak belirlenebilir. 3d max studio programı ile aynı ortam hazırlanıp ışın izleme ve ışın etkinlik metotları uygulandığında elde edilen görüntüler şekil 6.18'da 3d max studio programında ışın izleme algoritmasının çalıştırılması ve şekil 6.19'da 3d max studio programında ışın etkinlik algoritmasının çalıştırılması algoritmasının çalıştırılması gösterilmiştir.



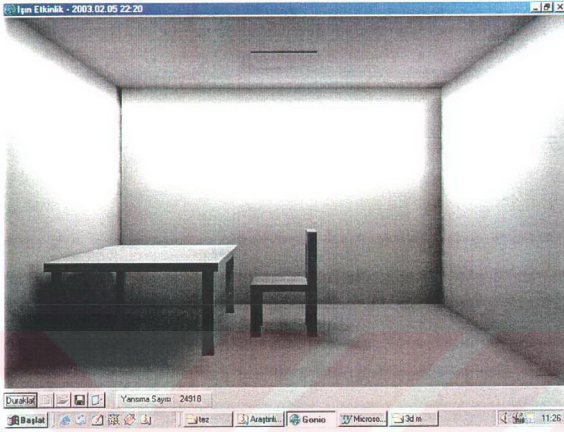
Şekil 6.18. 3d max studio programında ışın izleme algoritmasının çalıştırılması

3d max studio programında ışın izleme ve ışın etkinlik değerleri başlangıç parametreleri % 80 olarak ayarlanmıştır. Geliştirilen yazılımda kullanılan ışın izleme ve ışın etkinlik yansıma sayıları bakımından karşılaştırılmıştır.

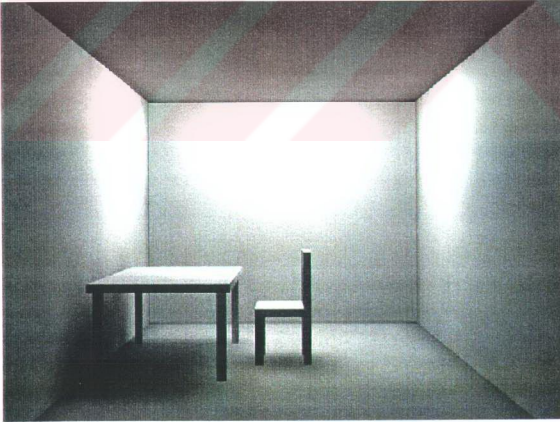


Şekil 6.19. 3d max studio programında ışın etkinlik algoritmasının çalıştırılması

3d max studio programı ile geliştirilen yazılım karşılaştırılarak aktarma metodlarının yansıma sayılarının belirlenmiştir. Aynı sanal ortam ile yapılan aydınlatma uygulamaları incelenmiştir. Bu incelemeler sonucunda elde edilen sonuçlar; Bu test sonucunda 3d max studio programında kullanılan ışık kaynağı ile geliştirilen yazılımda kullanılan ışık kaynağı benzerdir. Fakat aynı ışık kaynakları değildir. Bu durum geliştirilen yazılım bir üstünlüğünü ortaya koyar. Geliştirilen yazılım goniofotometre ile elde edilen verilerle çalıştırılmaktadır. Kullanılan verilerdeki ışık şiddet vektörleri büyük değerden azalarak küçülme gözlenirken, 3d max studio programında kullanılan ışık kaynağı sabit değerden ani azalmalarla net ve keskin inişler yapmaktadır. Bu nedenle oluşturulan şekillerde farklılıklar ortaya çıkmıştır. Şekil 6.20'de geliştirilen yazılımla 1607 yansıma sonunda elde edilen ışın etkinlik uygulaması ve şekil 6.21'de 3d max studio programıyla yapılan ışın etkinlik uygulaması gösterilmiştir.



Şekil 6.20. Geliştirilen yazılımla 1607 yansıma sonunda elde edilen ışın etkinlik uygulaması



Şekil 6.21. 3d max studio programıyla yapılan ışın etkinlik uygulaması

Bu sistem ile görüntü oluşturulurken kullanıcının en iyi ve en hızlı bir şekilde ideal şekle ulaşması için gerekli yansıma sayısını bilinmesi gerekmektedir. Yansıma sayısı artırılınca özellikle yumuşak gölgelerdeki değişimin azalarak göz ile tespit edilemeyecek kadar küçüldüğü tespit edilmiştir. Gerçekte sonsuz sayıda olan yansıma değeri programda iki alan arasındaki form faktörü değeri baz alınarak yansımalar sınırlandırılmıştır. Programda iki alan arasındaki form faktörü değeri başlangıç değerine göre karşılaştırılır. Başlangıç değeriyle son değer oranı binde bire ulaştığında yansımalar durdurulur.

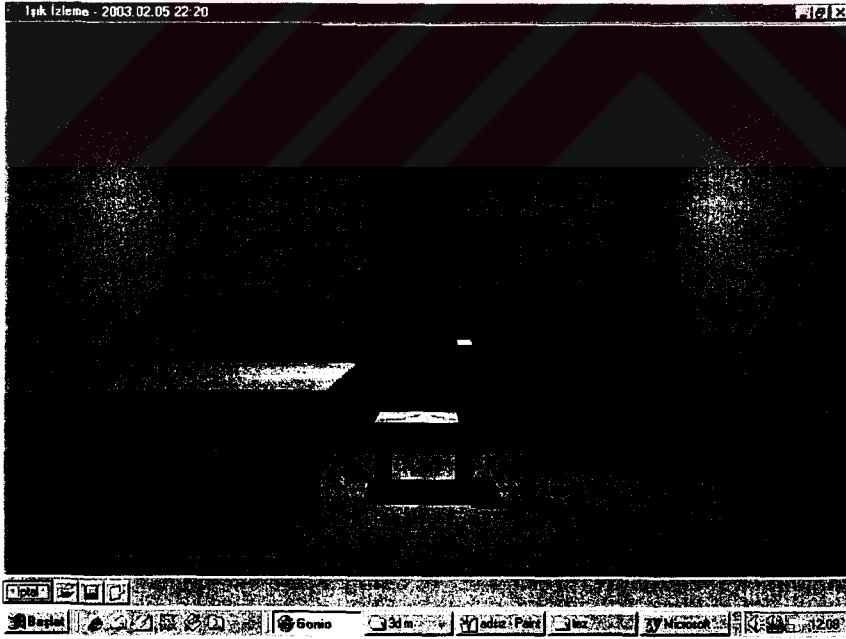
Intel Pentium-III 733 Mhz bir bilgisayar ile 175 saniyede alanlardan yayılan form faktörü değerinin 1607 yansımada başlangıç değerinin binde birlik değerine ulaşarak program yansımaları durdurmuştur.

3d max studio programındaki ışın izleme uygulamasıyla karşılaştırıldığında geliştirilen ışın izleme uygulamaları keskin gölgeler ve kısıtlı bir gerçeklik sunduğu görülmektedir. Sadece bakış yönü doğrultusuna gelen ışınların takip edilmesi kamera ve odak noktalarının sabit noktalar haline gelmesine sebep olur. Bu yüzden az sayıda ışın takibi gölgeleri olduğundan daha keskin ve koyu olarak tanımlar. Bu durum gerçeğe yakın görüntüler elde edilmesini sınırlamaktadır. Sadece bir ışık şiddeti eğrisi ile temsil edilebilen armatürlerin oluşturdukları görüntülerde ideale yaklaşılr. Bakış yönü doğrultusuna gelen az sayıda ışınların takip edilmesi sistemin ideal görüntü oluşturma hızını artırır.

Şekil 6.22’de geliştirilen yazılımla 580 yansıma sonucunda elde edilen ışın izleme uygulaması gösterilmiştir. Bu yapıda ışık kaynağından çıkan ışık şiddeti değeri tüm yansımalarını yapıp tüm alana dağılacak kadar yansıma yapmamıştır. Yansıma sayısı artırılarak bilimsel bir alt yapıyla form faktörü değerine göre yapılırsa Şekil 6.23’de geliştirilen yazılımla 1607 yansıma sonucunda elde edilen ışın izleme uygulaması gösterilmiştir. 1607 yansımada başlangıç değerine göre binde birlik form faktörü oluşmuştur. Bu form değeri yansımaları otomatik olarak durdurmuştur.

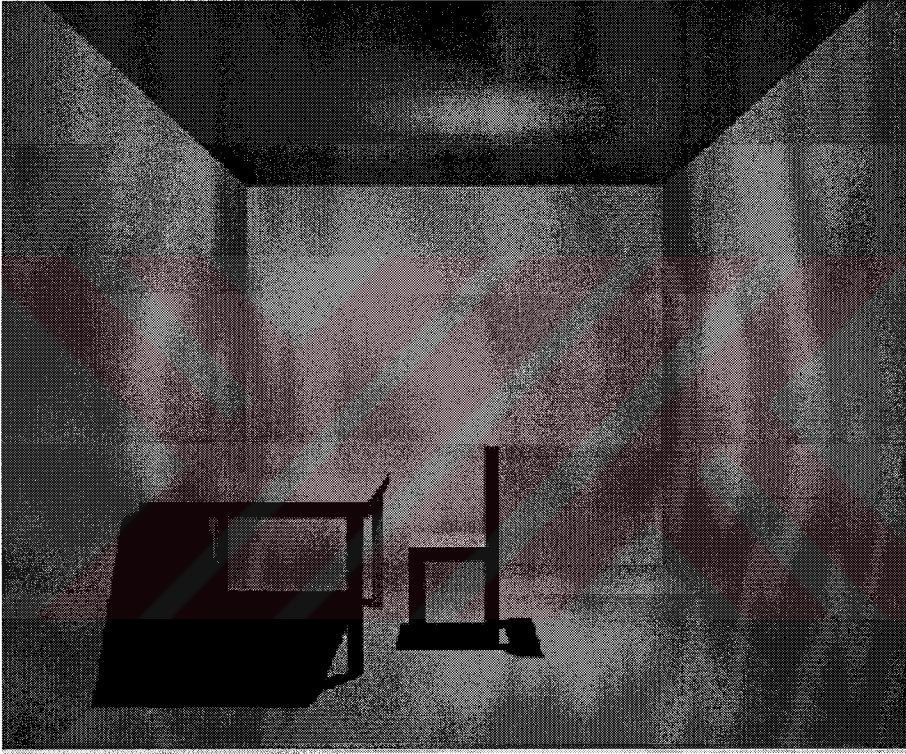


Şekil 6.22. Geliştirilen yazılımla 580 yansıma sonucunda elde edilen ışın izleme uygulaması



Şekil 6.23. Geliştirilen yazılımla 1607 yansıma sonucunda elde edilen ışın izleme uygulaması

Şekil 6.24’de 3d studio max programıyla yapılan ışın izleme uygulaması gösterilmiştir. 3d studio max programıyla yapılan ışın izleme uygulaması ile geliştirilen yazılımın ışın izleme uygulaması aynı sanal mekan kullanılarak karşılaştırılmıştır. 3d studio programının kısıtlı yapısından dolayı aynı ışık kaynağı uygulanamamıştır. Karşılaştırma ile elde edilen sonuçlar şunlardır; 3d studio max programıyla yapılan ışın izleme uygulaması daha keskin gölgeler oluşturur.



Şekil 6.24. 3d studio max programıyla yapılan ışın izleme uygulaması

7. SONUÇ

Geliştirilen goniofotometre armatürlerin ışık şiddeti eğrilerini çıkartan diğer goniofotometre modelleriyle karşılaştırıldığında özgün bir goniofotometre olma özelliği sunmaktadır.

Çalışmadan elde edilen sonuçlar ile kaynaklardan elde edilen sonuçlar karşılaştırıldığında, bilgisayarlı görme, görüntü sentezi gibi bilgisayar mühendisliği konularına fotometrik bir alt yapı kazandıran sistem kurulmuştur. Bu sistemin sağladığı fotometrik bilgiler mevcut bilgisayar teknolojileriyle geliştirilen görüntülerin doğruluğunu artırmaktadır.

Işın izleme ve ışın etkinlik metotlarının algoritmalarının farklı çalışmasından kaynaklanan görüntü farklılıklarının oluşması yazılım içerisinde iki ayrı görüntü aktarma metodunun kullanılmasını zorunlu kılmıştır. Her iki metodun da avantaj ve dezavantajları vardır.

Işın izleme metodu, güçlü ve bir ışık şiddeti eğrisi ile temsil edilebilen ışık kaynakları için kullanılması daha uygundur. Aynasal yansıma gösterdiğinden düzgün ve parlak yüzeyler tercih edilmelidir. Algoritmaları ışığın kırılma etkileri üzerine olduğu için yansıyan yüzeylerin renk değişimlerini gösterebilir. Sadece göze gelen ışınları takip eder, az sayıda ışın takibi gölgeleri olduğundan daha keskin ve koyu olarak tanımlar. Hızlı bir algoritma olması bu metodun önemli bir özelliğidir.

Işın etkinlik metodu, tüm ışık kaynaklarını modelleyebilir. Fakat ışın izlemeye göre yayınık dağılımlı ışık kaynakları olan iki ve daha fazla ışık şiddeti eğrisi ile temsil edilebilen ışık kaynaklarının benzetimi için uygundur. Dağınık sistemlerin alt alanlarını topladığından renk karışımlarını ve renk solmalarını gösterebilir. Yansıma ve dağılma üzerine kurulu algoritması yumuşak gölgeleri gösterebilir. Gölge tonlarını görebilme daha gerçekçi bir görüntü oluşmasını sağlar. Sanal alan içerisindeki tüm nesne ve ışık kaynakları ele alınarak herbirinin tek tek ele alınarak hesaplandığından bakış açısından bağımsızdır. Ayarlar penceresi ile koordinat

sistemi içinde X, Y, Z ekseninde odak, kamera ve ışık kaynaklarının yeri değiştirilerek görüntü oluşturulabilir. Örnek alandaki tüm nesneler ve birbirleriyle etkileşimleri ele alındığından yavaş bir algoritmadır.

Işık şiddeti eğrilerini çıkartan bu sistem ile piyasada bulunan çeşitli firmaların lamba ve armatür tipleri ile yetinmeyen, yeni ve özgün aydınlatma programlarına entegre ederek farklı tasarımlar üzerinde çalışan aydınlatma tasarımcıları armatür tasarımı ve geliştirilmesinde bu sistemi kullanılabılırler.

Aydınlatma derslerinde ışık şiddet eğrileri ile yapılacak olan deneyler; temel aydınlatma kavramları, armatür-renk ilişkisi, ışık-renk ilişkisi, aydınlık düzeyi hesapları, ışık kaynakları, ışık üretiminin temelleri, yararlanma katsayıları, aydınlatma verim hesapları gibi konularla ilişkilidir. Sistemin programına yapılacak olan yeni düzenlemelerle, yukarıdaki konuların eğitimi ve öğretimi kolaylaştırılarak hızlandırılacaktır. Kurulmuş olan sistem fotometrik çalışmalara destek olacak sanal fotometri laboratuarına temel sağlayabilecek düzeydedir.

Bu sistemle çok amaçlı uygulama armatürünün yapılması planlanmaktadır. Bu armatür bilgisayar kontrollü reflektörleri sayesinde aydınlatılacak hedef alanın istenilen alanına istenilen aydınlık düzeyini sağlar. Bu işlem için aydınlatması yapılacak olan alan sanal ortam çizilir. Ortamın herhangi bir alanına düşürülmesi planlanan aydınlık düzeyi için armatür denkleminin C katsayılarını yazılım belirler. Bu katsayılara göre reflektörler gerekli konuma gelir ve ışık kaynağının lambasına gerekli akım uygulanır. Böylece tiyatro salonu, gösteri merkezleri, televizyon stüdyoları gibi yerlerde yazılım kullanılabılır.

Işın izleme ve ışın etkinlik aktarma metotları üç boyutlu sanal ortamlarda armatürün ışık şiddet verilerine göre piksellerin yazılım değerlerini belirlemektedirler. Bu algoritmaları gerçek dijital fotoğrafın belirli bir noktasındaki piksel değerini okutarak gerçek görüntüden sanal ortamdaki derleme sayısını belirleme yolu seçilmelidir. Tecrübeler yaklaşık 24000 derleme sayısı yeterli olduğu görülmüştür. Bu rakamın üzerinde ne kadar derleme gerçekleşirse gerçekleşsin oluşan görüntüde gözle

algılanabilecek farklılıklar gözlenmemektedir. Bir nesne üzerindeki renk tonu görüntüsü aydınlık seviyesine bağlıdır. Kurulan sistem renk yansımalarını dikkate almaktadır. Nesnelerin gerçek renklerinde görüntülenebilmesi için aydınlatılmasında kullanılacak olan en uygun armatürün seçimini yine dijital resmin piksel değeri kullanılabilir.

Bu konuda çalışacakların özellikle birbirine paralel olarak çalışan çift işlemcili bilgisayarlarda ekran kartları yüksek çözünürlükte donanıma sahip olması önerilir. Goniofotometre tasarımlarında kullanılan fotometre başının hata payı %0,5 değerinin altında olmalıdır.

Işık şiddeti verileri ile dağılım alanları ve sınırları belirlenmiş olan armatürler aydınlatılacak yerin o andaki amacına göre bilgisayar programına seçtirilerek mekan içerisinde kullanılabilir.

Sistemin armatürlerin matematiksel modelini temel alarak görselleştirmesi mekanlara göre ışık kaynağı tasarımının yazılımla hesaplanabilmesine ortam sağlayacaktır. Yazılımda mekanın hangi noktalarında ne kadar aydınlık düzeyi olması gerektiği belirlenir. Yazılıma o alan için en optimum denklemleri hesaplatılır. Üç boyutlu ışık dağılım yüzeyi çizdirilebilir. Bu sistemle bir firmanın armatürlerinin goniofotometrik ölçümleri sonucunda denklemlerini elde edebilir. Bu denklemlerden en uygununu yapay zeka veya genetik gibi bir kontrol yöntemiyle hesaplatarak sistemin kendi kendine seçim yapmasını sağlayabilir.

Sistem ile armatürler ve sanal mekan matematiksel olarak tanımlanmış, ışığın mekan içi dağılımlarını sağlayan algoritmalar bir matematiksel model olarak ele alınmıştır. Mekanlarda kullanılan pürüzlü duvar, alçıpan, laminant kaplama vb. malzemelerin yansıtma ve yutma katsayıları dikkate alınmalıdır. Bu katsayıları belirlerken Goniofotometre kullanılabilir.

Sistem elektromanyetik tayf içerisinde bulunan tüm ışınlam çeşitleri için kullanılabilir. Sistem yalnızca ışık bölümünün şiddet dağılımı için kullanılmaktadır.

Fotometre başında yapılacak olan algılayıcı değişiklikle diğer ışınım enerjilerininide algılayabilir. Örneğin fotometre başlığı kızıl ötesi şiddet dağılımını algılamasıyla PIR ve AFIR algılayıcılarının algılama alanlarının belirlenmesi, radyo dalgalarının yayılma alanlarının belirlenmesi, baz istasyonlarının yayılım alanlarının belirlenmesi, radyoaktif malzemelerin yayılım alanlarının belirlenmesinde, flouresan lamba, kompak flouresan lamba gibi lambaların 380 ile 780 nm aralığında yaydıkları ışıktan başka insan sağlığını tehdit ışımalarının ölçülmesi gibi bir çok alanda kullanılabilir. Sistem lambaların verimliliklerinin karşılaştırılması için bir test aracı olarak kullanılabilir.



KAYNAKLAR

1. CIE, "The Measurement Of Absolute Luminous Intensity Distributions", *CIE Publications*, No: 70, Vienna, 7-27 (1987).
2. Bal, G. ve Bekiroğlu, E., "Design and implementation of digitally controlled step motor drives for toothbrush / dentifrice mechanism", *International Conference On Electrical Machines*, Istanbul, 2: 803-807 (1998).
3. Şerefhanoglu, M., "Aydınlatma tasarımında mimarın ve elektrik mühendisinin rolü" TMMOB Elektrik Mühendisleri Odası, *II. Ulusal Aydınlatma Sempozyumu*, Diyarbakır, 1-2 (2003).
4. Amantides J., "Ray tracing with cones", *Computer Graphics (SIGGRAPH '84 Proceedings)*, 18(3): 129-135 (1984).
5. Whitted, T., "An improved illumination model for shaded display", *Communication of the ACM*, 23 (6): 343-349 (1980).
6. Cohen, M., Greenberg, D. P., Immel, D. S., Brock, P. J., "An efficient radiosity approach for realistic image synthesis", *IEEE Computer Graphics and Applications*, 6(3): 26-35 (1984).
7. Cook, R. L., "Stochastic sampling in computer graphics", *ACM Transactions on Graphics*, 5:1 51-57 (1986).
8. Blinn, J. F., "Models of light reflection for computer synthesized pictures", *Computer Graphics (SIGGRAPH '77 Proceedings)*, 11:2 192-198 (1977).
9. Cook, R. L., Torrance, K. H., "A reflection model for computer graphics", *ACM Transactions on Graphics*, 1:1 7-24 (1982).
10. Cook, R. L., Porter, T., Carpenter, L., "Distributed ray tracing", *Computer Graphics, (SIGGRAPH '84 Proceedings)*, 18(3): 137-145 (1984).
11. Phong, B. T., "Illumination for computer generated pictures", *Communication of the ACM*, 18(6): 311-317 (1975).
12. Siegel, R., Howell, J. R., "Thermal Radiation Heat Transfer", 3rd Edition. *Hemisphere Publishing Corporation*, New York, 7-11 (1992).

13. Nishita, T., Nakamae, E., "Continues Tone Representation Of Three Dimensional Objects Taking Account of Shadows and Interrefraction", *ACM Siggraph Computer Graphics*, 19(2): 23-30 (1985).
14. Cohen, M., Chen, S. E., Wallace, J. R., Greenberg, D. P., "A progressive refinement approach to fast radiosity image generation", *Computer Graphics (SIGGRAPH '88 Proceedings)*, 22(4): 75-84 (1988).
15. Hanrahan, P., Salzman, D., "A rapid hierarchical radiosity algorithm", *ACC siggraph computer graphics*, 24(4): 197-206 (1991).
16. Appel, A., "Some techniques for shading machine renderings of solids", *Proceeding the Spring Joint Computer Conference*, 6(1): 85-103 (1985).
17. Lighting Technologies, Inc., "*FIELD, Finite Element Luminaire Design*", *Version 2.1 User's Guide. Boulder, CO: Lighting Technologies, Inc*, New York, 5-7 (1990).
18. Carbonano, G., "3d max yaratıcı teknikler", Çeviri, Ali Halaç, *Sistem Yayıncılık*, İstanbul, 52-56 (1999).
19. IES, "IES approved method for photometric measuring and reporting tests on reflector type lamps", *Journal of the IES*, New York, 2-3 (1997).
20. IES, "IES Lighting Handbook", 2. Baskı, *IES Publications*, NewYork, 17-19 (1987).
21. IES, "IES Lighting Handbook", 2. Baskı, *IES Publications*, NewYork, 7-8. (1981).
22. IES, "IES approved method for electrical and photometric measurement of general service incandescent lamps", *Journal of the IES*, New York, 2-6 (1982).
23. IES, "IES approved method for electrical and photometric measurement of high intensity discharge lamps", *Journal of the IES*, New York, 21-24 (1988).
24. IES, "IES general guide to photometry", *Journal of the IES*, New York, 12-14 (1997).
25. Özkaya, M., "Aydınlatma Tekniği", 7. Baskı, *Birsan Yayınevi*, İstanbul, 56 (1997).
26. Sutherland, I. E., "A man-machine graphical communication system", *In Proceedings of the Spring Joint Computer Conference*, London, 6(1): 7-8 (1963).

27. Bounight, W. J., "A procedure for generation of three-dimensional half-tone computer graphics presentations", *Communications of the ACM*, 13:9 292-301 (1970).
28. Gouraud, H., "Computer display of curved surfaces", *Tech. Rep. Dept. Of Computer science, University of Utah*, Salt Lake City, Utah, 8-9 (1971).
29. Nazlı, H., "Radiosity", *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Yüksek Lisans Tezi, İstanbul, 25-86 (1997).
30. Sillicon F., "Radiosity Global Illumination", *Morgan Kaufmann Publishers Inc*, San francisco, 12-23 (1991).
31. Ashdown, I., "Radiosity: a programmer's perspective", *John Wiley & Sons, Inc*, NewYork, 112-123 (1994).
32. Chon, M. F., Wallace J. R. "Radiosity And Realistic Image Synthesis", *Academic Press*, London, 32-36 (1993).
33. Max, N. L., Allison M.J., "Lineer radiosity approximation using vertex to vertex form factors graphics gems III", *IEE Computer Graphics and Application*, 8(3) 52-58 (1992).
34. Wolf, R.J., "3d Graphics A Visual Approach", *Oxford University Press*, Newyork, 56-86 (2000).
35. Cohen, M., Greenberg, D.P., Immel, D.S., Brock, P.J., "An Efficient Radiosity Approach For Realistic Image Synthesis", *IEE Computer Graphics and Application*, 74-79 (1985).
36. Qiulin, D., "Surface Enginnering Geometry For Computer-Aided Design And Manufacture", *John Wiley & Sons, Inc*, New york. 95-97 (1987).
37. Gürdal, O., "Algılayıcılar ve Dönüştürücüler", 2. Baskı, *Nobel yayınevi*, Ankara, 152-153 (2000).
38. Sirel, Ş., "Aydınlatma Sözlüğü", *Birsen Yayınevi*, İstanbul, 84-85 (1997).
39. Kenjo, T., "Stepping motors and Their Microprocessor Controls", *Clarendon Press, Oxford*, London, 181-188 (1984).

40. Tooley, M., "Newnes pocket book", *Butterworth heinemann ltd*, London, 34-36 (1992).
41. Üncü, İ.S., Gürdal, O., "Işık şiddet eğrilerinin bilgisayar yardımı ile çıkartılması", *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Yüksek lisans tezi, Ankara, 23-31 (1996).
42. Üncü, İ. S., Gürdal, O., "Measuring planes used for design of goniophotometer and a goniophotometer application", *Journal of the Institute of Science and Technology of Gazi University*, 15(1): 303-310 (2002).
43. Karagülle, İ., Pala, Z., " C++ Builder 6.0", *Türkmen Yayıncılık*, İstanbul, 68-584 (2003).
44. İnan, Y., Demirci, N., " C++ Builder 6", *Prestige Yayıncılık*, İstanbul, 400-412 (2004).
45. Gouraud, H., "Computer display of curved surfaces", *Tech. Rep. Dept. Of Computer Science, University of Utah*, Salt Lake City, Utah, 41-56 (1971).
46. Goral, C. M., Torrance, K. E., Greenberg, D. P., Battaile, B., "Modelling the interaction of light between diffuse surfaces", *Computer Graphics (SIGGRAPH '84 Proceedings)* 18(3): 212-222 (1984).
47. Michael F. C., John R. W., "Radiosity and Realistic Image Synthesis", *Computer Graphics (SIGGRAPH '93 Proceedings)*, 1(2): 4-8 (1993).
48. Akın, Ö., "Nümerik analiz", *Ankara Üniv.Fen Fakültesi Ders Kitapları*, 158-164 (1998).
49. Krishnamurthy, E.V., Sen, S.K., "Numerical Algorithms", *Affiliated East West Press Private limited*, New delphi, 232-234, (1996).
50. Roger, F. D., Adams, A. J., "Mathematical elements for computer graphics", *McGraw-Hill Publishing Company*, New York, 107-110 (1989).



EK 1. Yazılımı Oluşturan Alt Pencerelelerin Kodları

//IŞIK AYARLARI KODLARI

//-----

#ifndef IIAyalariH

#define IIAyalariH

//-----

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <ComCtrls.hpp>

#include "LMDBaseEdit.hpp"

#include "LMDCustomBevelPanel.hpp"

#include "LMDCustomControl.hpp"

#include "LMDCustomEdit.hpp"

#include "LMDCustomMaskEdit.hpp"

#include "LMDCustomPanel.hpp"

#include "LMDMaskEdit.hpp"

#include "IIPen.h"

//-----

class TIIAyarlar : public TForm

{

__published: // IDE-managed Components

TGroupBox *GroupBox1;

TLabel *Label1;

TLMDMaskEdit *IX1;

TUpDown *UpDown1;

```

TLabel *Label2;
TLMDMaskEdit *IY2;
TUpDown *UpDown2;
TLabel *Label3;
TLMDMaskEdit *IZ3;
TUpDown *UpDown3;
TGroupBox *GroupBox2;
TLabel *Label4;
TLabel *Label5;
TLabel *Label6;
TLMDMaskEdit *KX4;
TUpDown *UpDown4;
TLMDMaskEdit *KY5;
TUpDown *UpDown5;
TLMDMaskEdit *KZ6;
TUpDown *UpDown6;
TGroupBox *GroupBox3;
TLabel *Label7;
TLabel *Label8;
TLabel *Label9;
TLMDMaskEdit *OX7;
TUpDown *UpDown7;
TLMDMaskEdit *OY8;
TUpDown *UpDown8;
TLMDMaskEdit *OZ9;
TUpDown *UpDown9;
void __fastcall AyarTusuBasildi(TObject *Sender, bool &AllowChange,
    short NewValue, TUpDownDirection Direction);
void __fastcall FormActivate(TObject *Sender);
void __fastcall FormDeactivate(TObject *Sender);
private:      // User declarations
public:       // User declarations

```

```

__fastcall TIIAyarlar(TComponent* Owner);
class TIsikIzlemePen * II;
};
//-----
extern PACKAGE TIIAyarlar *IIAyarlar;
//-----
#endif

// IŞIK KAYNAĞI MODELİ KODLARI

#ifndef ISIK_H
#define ISIK_H

#include "light.h"
#include "stdlib.h"
#include "IKModeli.h"

class Isik : public Light {
public:
    Isik( IKModeli & IKM ) {
        IK = & IKM;
    }

    Light *clone() const
    { return new Isik(*this); }

    Intensity computeIntensity(const Matrix41& point) const;

private:
    IKModeli * IK;
};

```

```
#endif
```

```
//-----
```

```
#ifndef IIDRLYC_H
```

```
#define IIDRLYC_H
```

```
//-----
```

```
#include <Classes.hpp>
```

```
#include "raytracershow.h"
```

```
//-----
```

```
class IsikIzlemeIsi : public TThread
```

```
{
```

```
protected:
```

```
    void __fastcall Execute();
```

```
public:
```

```
    __fastcall IsikIzlemeIsi();
```

```
    void Basla( RaytracerShow & );
```

```
    void Bittir();
```

```
    void ( __closure * OnOver )();
```

```
private:
```

```
    bool render;
```

```
    RaytracerShow * RTS;
```

```
};
```

```
//-----
```

```
#endif
```

```
#include "IKDenk.h"
```

```
#define MIN(a, b) (((a) < (b)) ? (a) : (b))
```

```
void IKDenk::PlyHspl( mxArray **p, int b )
```

```
{
```

```
    mxArray *y=mclGetUninitializedArray(),
```

```
        *x=mclGetUninitializedArray(),
```

```
        *z=mclGetUninitializedArray(),
```

```
        *s;
```

```
    double *d, *f;
```

```
    y = mxCreateDoubleMatrix( 1, sutun+2, mxREAL); // Eldeki verilerden polinom
oluşturacak
```

```
    d = mxGetPr(y);
```

```
    z = mxCreateDoubleMatrix( 1, satir, mxREAL); // Eldeki verilerden polinom
oluşturacak
```

```
    f = mxGetPr(z);
```

```
    for( int a=0; a<sutun; a++ ){
```

```
        for( int c=0; c<satir; c++ )
```

```
            f[c] = deger[ a + c * sutun ];
```

```
    s = mclGetUninitializedArray();
```

```
    mlfAssign(&s, mlfPolyval(NULL,z, mlfScalar(b),NULL));
```

```
    d[a+1]= mxGetPr(s)[0];
```

```
        mxDestroyArray(s);
```

```
}
```

```
d[0] = d[2];
```

```
// önceki ve sonraki veri devamlılığ
```

```
d[sutun+1] = d[sutun-1];
```

```
mxDestroyArray(x);
```

```

if( sutun > 1 ){
    double acar = 90.0 / ( sutun - 1 );
    mlfAssign(&x, mlfColon(mlfScalar(-acar), mlfScalar(acar), mlfScalar( 90 +
acar ))); // yeni de erlerle enlemleri olu tur
} else
    mlfAssign(&x, mlfColon(mlfScalar(-90), mlfScalar(90), mlfScalar(90)));

```

```

mlfAssign(p, mlfPolyfit(NULL,x,y,mlfScalar(MIN(6,sutun+1)))); // polinom
olu tur

```

```

mxDestroyArray(y);
mxDestroyArray(x);
mxDestroyArray(z);
}

```

```

void IKDenk::operator = ( IKModeli & IK )
{
    if( !sutun || !satir )
        return;

```

```

double acix = 90.0 / (sutun-1),
        aciy = 90.0 / (satir-1);
mxArray *x=mclGetUninitializedArray(),
        *y=mclGetUninitializedArray(),
        *p=mclGetUninitializedArray();
mlfAssign(&x, mlfColon(mlfScalar(0), mlfScalar(aciy), mlfScalar( 90+aciy ))); //
yeni de erlerle enlemleri olu tur
y = mxCreateDoubleMatrix( 1, satir+1, mxREAL); // Eldeki verilerden polinom
olu turacak
double *d = mxGetPr(y);

for( int a=0; a<sutun; a++ ){

```

```

        for( int b=0,c=0; b<satir; b++ )
            d[c++] = IK.IsikSiddeti( a*acix, b*aciy );
    d[satir] = d[satir-2];

```

```

        mlfAssign(&p, mlfPolyfit(NULL,x,y,mlfScalar(MIN(6,satir-1)))); //

```

polinom oluþtur

```

    double *t = mxGetPr(p);

```

```

    for( int b=satir-1; b>=0; b-- )
        deger[a+b*sutun] = t[b];
    mxDestroyArray(p);

```

```

}

```

```

mxDestroyArray(x);

```

```

mxDestroyArray(y);

```

```

    degerno++;

```

```

}

```

```

#ifdef matlab_h

```

```

#define matlab_h

```

```

#ifdef __cplusplus

```

```

    extern "C" {

```

```

#endif

```

```

#include <stddef.h>

```

```

#ifdef _H_STANDARDS && !defined(__cplusplus)

```

```

#undef NULL

```

```

#define NULL ((void *) 0L)
#endif

#include "matrix.h"
#if defined(MLF_V1_2) && defined(MLF_V2)
#undef MLF_V2
#endif

#if !defined(MATLAB_COMPILER_GENERATED_CODE) &&
!defined(MLF_V1_2) && !defined(MLF_V2)
#define MLF_V2
#endif

#ifdef MLF_V2
#define MLF_ENABLE_ND_ARRAYS 1
#define MLF_USE_VARARGOUT 1
#ifndef __cplusplus
#define MLF_ENABLE_TRYCATCH 1
#endif /* __cplusplus */
#endif /* MLF_V2 */

#include "libmatlb.h"
#include "libmmfile.h"
#include "libsgl.h"
#include "libsglm.h"

#ifndef MLF_ENABLE_ND_ARRAYS
#define mlfOnes(A,B) mlfOnes(A, B, NULL)
#define mlfRand(A,B) mlfRand(A, B, NULL)
#define mlfRandn(A,B) mlfRandn(A, B, NULL)
#define mlfZeros(A,B) mlfZeros(A, B, NULL)
extern mxArray * mlfSize2d(mxArray **OO1, mxArray *RI1, mxArray *OI1);
#define mlfSize mlfSize2d
#endif

#if !defined(MLF_V2)
#if !defined(MATLAB_COMPILER_GENERATED_CODE)

```

```

#define mlfDel2(A,B,C) mlfDel2(A, B, C, NULL)
#define mlfDatevec(A,B,C,D,E,F) mlfDatevec(A,B,C,D,E,F,NULL)
#define mlfQr(A,B,C,D) mlfQr(A,B,C,D,NULL)
#define mlfLu(A,B,C) mlfLu(A,B,C,NULL)
#define mlfAutomesh(A,B,C) mlfAutomesh(A,B,C,NULL)
#define mlfCov(A,B,C) mlfCov(A,B,C,NULL)
#define mlfStrjust(A) mlfStrjust(A,NULL)
#define mlfNumjac(o2,o3,i1,i2,i3,i4,i5,i6,i7,i8,i9) \
    mlfNumjac(o2,o3,NULL,i1,i2,i3,i4,i5,i6,i7,i8,i9,NULL)
#define mlfFmin(A,B,C,D,E,F) mlfFmin(A,B,C,D,E,F,NULL)
#define mlfFmins(A,B,C,D,E,F) mlfFmins(A,B,C,D,E,F,NULL)
#define mlfFzero(A,B,C,D,E) mlfFzero(NULL, NULL, NULL, A, B, C, D, E,
NULL)
#define mlfIcubic(A,B,C) mlfIcubic(A,B,C,NULL)
#define mlfInterp1(A,B,C,D) mlfInterp1(A,B,C,D,NULL)
#define mlfInterp2(A,B,C,D,E,F) mlfInterp2(A,B,C,D,E,F,NULL)
#define mlfInterp4(A,B,C,D,E) mlfInterp4(A,B,C,D,E,NULL)
#define mlfInterp5(A,B,C,D,E) mlfInterp5(A,B,C,D,E,NULL)
#define mlfInterp6(A,B,C,D,E) mlfInterp6(A,B,C,D,E,NULL)
#define mlfQuad(out,A,B,C,D,E,F) mlfQuad(out,A,B,C,D,E,F,NULL)
#define mlfQuad8(out,A,B,C,D,E,F) mlfQuad8(out,A,B,C,D,E,F,NULL)
#endif /*!defined(MATLAB_COMPILER_GENERATED_CODE) */
#define mlfOde113 mlfOde113_v4
#define mlfOde15s mlfOde15s_v4
#define mlfOde23 mlfOde23_v4
#define mlfOde23s mlfOde23s_v4
#define mlfOde45 mlfOde45_v4
#define mlfOdeget mlfOdeget_v4
#define mlfOdeset mlfOdeset_v4
#define mlfGradient mlfGrd_v4
#endif

```

```

extern void mlfHGInitialize(int *argc, const char *argv[]);
extern void mlfHGWaitForFiguresToDie();
extern void mlfHGTerminate();
extern mxArray * mxColonop ( double, double, double );
typedef void (*mlfFuncp)( void );
typedef int (*mlfThunkp)( mlfFuncp pFunc, int nlhs, mxArray **plhs, int nrhs,
mxArray **prhs );
typedef struct {
    char *name;
    mlfFuncp pFunc;
    mlfThunkp pThunk;
} mlfFuncTabEnt;
typedef mlfFuncTabEnt mlfFuncTab; /* Table is just an array of entries */
typedef void (*mlxFunctionPtr)(int nlhs, mxArray * plhs[], int nrhs, mxArray *
prhs[]);
typedef struct mlfFunctionTableEntry_tag {
    const char *name;
    mlxFunctionPtr mlxFcn;
    int nargin;
    int nargout;
} mlfFunctionTableEntry;
extern void mlfFunctionTableSetup(size_t size, mlfFunctionTableEntry *pTab);
extern void mlfFunctionTableTakedown(size_t size, mlfFunctionTableEntry *pTab);
extern mlxFunctionPtr mclFevalLookup(mxArray* fcn, size_t size,
                                   mlfFunctionTableEntry *pTable);
extern mxArray *mclNargoutLookup(mxArray* fcn, size_t size,
                                   mlfFunctionTableEntry *pTable);
extern mxArray *mclNarginLookup(mxArray* fcn, size_t size,
                                   mlfFunctionTableEntry *pTable);
extern mlxFunctionPtr mlfFevalLookup(mxArray *fcn);

```

```

extern void mlfFeval ( int nlhs, mxArray **plhs, int nrhs, mxArray **prhs, char
*name );
extern void mlfFevalTableSetup ( mlfFuncTab *mlfUfuncTable );
extern void mlxCellfun(int nlhs, mxArray * plhs[], int nrhs, mxArray * prhs[]);
extern mxArray *mlfCellfun(mxArray *fun, mxArray *c, mxArray *k);
extern void mlxEomday(int nlhs, mxArray * plhs[], int nrhs, mxArray * prhs[]);
extern mxArray * mlfEomday(mxArray * y, mxArray * m);
extern void mlxDatenum(int nlhs, mxArray * plhs[], int nrhs, mxArray * prhs[]);
extern mxArray * mlfDatenum(mxArray * y,
                        mxArray * mo,
                        mxArray * d,
                        mxArray * h,
                        mxArray * mi,
                        mxArray * s);
extern mxArray * mlfDvcore(mxArray * t);
extern void mlxDvcore(int nlhs, mxArray * plhs[], int nrhs, mxArray * prhs[]);
extern mxArray * mlfSort(mxArray * * i, mxArray * x, mxArray * dim);
extern void mlxSort(int nlhs, mxArray * plhs[], int nrhs, mxArray * prhs[]);
extern mxArray * mlfExist(mxArray *in1, mxArray *in2);
extern void mlxExist(int nlhs, mxArray *plhs[], int nrhs, mxArray *prhs[]);
extern void mlxIsmembc(int nlhs, mxArray * plhs[], int nrhs, mxArray * prhs[]);
extern mxArray * mlfIsmembc(mxArray * x, mxArray * y);
#define mclCastToMxarray(x) (mxArray *) (x)
#ifdef MLF_USE_VARARGOUT
#define mlfFeval mlfVarargFeval
#endif

#ifdef __cplusplus
} /* extern "C" */
#endif

#endif /* matlab_h */

```

```
#ifndef IModelleyici_H
#define IModelleyici_H
```

```
//IŞIK MODELLEYİCİ KODLARI-----
```

```
-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Grids.hpp>
#include <Menus.hpp>
#include "IKModeli.h"
#include "LMDBaseEdit.hpp"
#include "LMDCustomBevelPanel.hpp"
#include "LMDCustomControl.hpp"
#include "LMDCustomEdit.hpp"
#include "LMDCustomMaskEdit.hpp"
#include "LMDCustomPanel.hpp"
#include "LMDMaskEdit.hpp"
#include <Buttons.hpp>
class TModelleyici : public TForm
{
__published: // IDE-managed Components
    TStringGrid *Veriler;
    TPopupMenu *PopupMenu;
    TMenuItem *SutunEkle1;
    TMenuItem *Sutunkar1;
    TMenuItem *N1;
    TMenuItem *SatrEkle1;
    TMenuItem *Satrkari;
    TLabel *Label1;
    TLMDDMaskEdit *IskAks;
```

```

TLabel *Label2;
TLMDMaskEdit *OlcUzklg;
TLabel *Label3;
TComboBox *LmbTr;
TBitBtn *Uygula;
void __fastcall SutunEkle(TObject *Sender);
void __fastcall SutunCikar(TObject *Sender);
void __fastcall SatirEkle(TObject *Sender);
void __fastcall SatirCikar(TObject *Sender);
void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
void __fastcall VerilerSetEditText(TObject *Sender, int ACol, int ARow,
    const AnsiString Value);
void __fastcall VerilerSelectCell(TObject *Sender, int ACol, int ARow,
    bool &CanSelect);
void __fastcall VerilerExit(TObject *Sender);
void __fastcall FormActivate(TObject *Sender);
void __fastcall UygulaClick(TObject *Sender);
protected:
    void TabloDegerleri();    // User declarations
    int OCol, ORow;
    bool dgat;
    IKModeli * IK;
    char adno,degerno;
public:    // User declarations
    __fastcall TIModelleyici(TComponent* Owner);
    __fastcall ~TIModelleyici();
    virtual void IKAta( IKModeli & );
};
#endif
// IŞIK ŞİDDET EĞRİSİ KODLARI
#include <vcl.h>
#pragma hdrstop

```

```

USERES("ISE.res");
USEFORM("Unit1.cpp", Form1);
USEFORM("ISEPen.cpp", ISEPen);
USELIB("..\\IsikKaynagiModeli\\_lib95841.lib");
USELIB("..\\IsikKaynagiModeli\\_lib95842.lib");
USELIB("..\\IsikKaynagiModeli\\_lib95843.lib");
USELIB("..\\IsikKaynagiModeli\\_lib95844.lib");
USELIB("..\\IsikKaynagiModeli\\_lib95845.lib");
USELIB("..\\IsikKaynagiModeli\\IKLib.lib");
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    return 0;
}
#ifdef ISEPen_H
#define ISEPen_H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <AppEvnts.hpp>

```

```

#include <Dialogs.hpp>
#include <ExtDlgs.hpp>
#include <Menus.hpp>
#include <ExtCtrls.hpp>
#include "OpenGLPanel.h"
#include "IKModeli.h"
class TISEPen : public TForm
{
    __published: // IDE-managed Components
        TButton *MenuButton;
        TPopupMenu *Menu;
        TMenuItem *N2Boyutlu;
            TMenuItem *N3Boyutlu1;
        TMenuItem *RYazdr1;
        TMenuItem *RKaydet1;
            TPrintDialog *PrintDialog1;
            TSavePictureDialog *SavePictureDialog1;
        TMenuItem *Kafes1;
        TMenuItem *Kapali1;
        TMenuItem *N2;
            TMenuItem *Renkli;
        TOpenGLPanel *ISgrafigi;
        TApplicationEvents *ApplicationEvents1;
        TScrollBar *YatayBar;
        TTimer *HintTimer;
            TMenuItem *N4;
        TScrollBar *DikeyBar;
        void __fastcall RYazdr1Click(TObject *Sender);
        void __fastcall RKaydet1Click(TObject *Sender);
        void __fastcall ISgrafigiInit(TObject *Sender);
        void __fastcall ISgrafigiResize(TObject *Sender);
        void __fastcall ISgrafigiPaint(TObject *Sender);

```

```

    void __fastcall FormResize(TObject *Sender);
void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
void __fastcall ISgrafigiMouseMove(TObject *Sender,
    TShiftState Shift, int X, int Y);
void __fastcall Kafes1Click(TObject *Sender);
void __fastcall Kapal1Click(TObject *Sender);
void __fastcall RenkliClick(TObject *Sender);
void __fastcall DikeyBarScroll(TObject *Sender, TScrollCode ScrollCode,
    int &ScrollPos);
void __fastcall HintTimerTimer(TObject *Sender);
void __fastcall N2BoyutluClick(TObject *Sender);
void __fastcall YatayBarScroll(TObject *Sender, TScrollCode ScrollCode,
    int &ScrollPos);
void __fastcall FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize);
void __fastcall FormActivate(TObject *Sender);
void __fastcall MenuButtonClick(TObject *Sender);
private:
    void BoyutDegis( bool )
#ifdef PSGorevH
#define PSGorevH
//-----
#include <Classes.hpp>
#include "comm.h"
#include <vector>
#include <string>
//-----
class TPSGorev : public TThread
{
protected:
    TCommPort m_CommPort;
    void __fastcall Execute();

```

```

    void __fastcall SynchronizeProgress(void);
public:
    __fastcall TPSGorev();
    virtual __fastcall ~TPSGorev();
    int Connect();
    void Disconnect();
    bool GetConnected()
    {
        return m_CommPort.GetConnected();
    }
    TCommPort& GetCommPort()
    {
        return m_CommPort;
    }
    int GFMoku(unsigned short yol, unsigned short dol );
    void GFMdur();
private:
    unsigned YatOlc;
    unsigned DikOlc;
    unsigned OlcVer;
    unsigned OlcNmr;
    unsigned YatAci;
    unsigned DikAci;
};
#endif

// PORT KODLARI

GLUQuadricObj *qobj;
int x, y,
    minHeight;
THintWindow *MyHint;

```

```

IKModeli * IK;
char adno, degerno;
public:      // User declarations
    __fastcall TISEPen(TComponent* Owner);
    __fastcall ~TISEPen();
    void IKAta( IKModeli & );
};
#endif
#ifndef PSPenH
#define PSPenH
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
class TPSPen : public TForm
{
    __published: // IDE-managed Components
        TLabel *Label2;
        TLabel *Label3;
        TButton *Baslat;
        TButton *Duraklat;
        TProgressBar *YTislmBari;
        TProgressBar *DTislmBari;
        TButton *Ayarlar;
private:      // User declarations
public:      // User declarations
    __fastcall TPSPen(TComponent* Owner);
};
//extern PACKAGE TPSPen *PSPen;
#endif
#endif PABH

```

```

#define PABH
#include "LptUnit.hpp"
#include "comm.h"
#include "MAPen.h"
#include "PSPen.h"
class PArabirimi{
public:
    PArabirimi();
    ~PArabirimi();

    void Ayarlar();
    void PortSurucu();
    void OlcSayilari( int, int );
    short int * Olcumler;
    void ( __closure * VeriAlimiTamamlandi)( int, int, short const * );
private:
    int PPortAc();
    int SPortAc();
    void __fastcall SaniyedeOnDefa(TObject *Sender);
    void __fastcall TusMouseUp(TObject *Sender, TMouseButton Button,
    TShiftState Shift, int X, int Y);
    void __fastcall TusMouseDown(TObject *Sender, TMouseButton Button,
    TShiftState Shift, int X, int Y);
    void __fastcall AyarlanabilirDegerlerExit(TObject *Sender);
    void __fastcall BaslatClick(TObject *Sender);
    void __fastcall DuraklatClick(TObject *Sender);
    void __fastcall AyarlarClick(TObject *Sender);
    void __fastcall PSPClose(TObject *Sender, TCloseAction &Action);
    int sn,mtr;
    int VeriAlimi;
    int YOSay, DOSay, BYO, BDO;
    int ydon, ddon;
    float YAdimAci, DAdimAci;

```

```

float AyarlanabilirDegerler[5];
    TTimer *Timer;
    TVicLPT *LPT;
    TCommPort *CMP;
    TMtrAyarPen *MAP;
    TPSPen * PSP;
};
#endif
//IŞIN ETKİNLİK KODLARI
#include <vcl.h>
#pragma hdrstop
USERES("rad.res");
USEFORM("Unit1.cpp", Form1);
USEUNIT("Triangle.cpp");
USEUNIT("IEDrlyc.cpp");
USEUNIT("JVector.cpp");
USEFORM("isinetknk.cpp", IsinEtkin);
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    return 0;
}

```

```

#include <vcl.h>
#include "pch.h"
#pragma hdrstop
#include "IEDrlyc.h"
#pragma package(smart_init)
IEDerleyici *IED;
__fastcall IEDerleyici::IEDerleyici(bool CreateSuspended)
    : TThread(CreateSuspended)
{
    Patches = new Triangle*[ 10000 ];
    MaxPtch = 10000;
    PatchCounter = 0;
    DIVISOR = 3;
    iterations = 0;

    Frms = new TList;
    Priority=tpLower;
    kilit = false;

    JVector T1,T2,T3,T4;
    // FLOOR
    T1.Edit( 1, -1, -1 );
    T2.Edit( -1, -1, -1 );
    T3.Edit( -1, -1, 1 );
    T4.Edit( 1, -1, 1 );
    IEDer->DivideQuad(T1, T2, T3, T4, 1, 0, 0, 0, 1, 1, 1);
    T4.Edit(1, 1, 1);
    T3.Edit(1, 1, -1);
    T2.Edit(-1, 1, -1);
    T1.Edit(-1, 1, 1);
    IEDer->DivideQuad(T1, T2, T3, T4, 2, 0, 0, 0, 1, 1, 1);

```

```
// BACK WALL
```

```
T1.Edit( 1, -1, 1 );
```

```
T2.Edit(-1, -1, 1 );
```

```
T3.Edit(-1, 1, 1 );
```

```
T4.Edit( 1, 1, 1 );
```

```
IEDer->DivideQuad(T1, T2, T3, T4, 3, 0, 0, 0, .7, .7, .7);
```

```
T4.Edit( 1, -1, -1 );
```

```
T3.Edit(-1, -1, -1 );
```

```
T2.Edit(-1, 1, -1 );
```

```
T1.Edit( 1, 1, -1 );
```

```
IEDer->DivideQuad(T1, T2, T3, T4, 3, 0, 0, 0, .7, .7, .7);
```

```
T1.Edit(-1, -1, 1);
```

```
T2.Edit(-1, -1, -1);
```

```
T3.Edit(-1, 1, -1);
```

```
T4.Edit(-1, 1, 1);
```

```
IEDer->DivideQuad(T1, T2, T3, T4, 4, 0, 0, 0, .23, .75, .3);
```

```
T1.Edit(1, -1, -1);
```

```
T2.Edit(1, -1, 1);
```

```
T3.Edit(1, 1, 1);
```

```
T4.Edit(1, 1, -1);
```

```
IEDer->DivideQuad(T1, T2, T3, T4, 5, 0, 0, 0, .88, .23, .3);
```

```
T1.Edit(.7, -1, .4 );
```

```
T2.Edit(.7, .2, .4 );
```

```
T3.Edit(.1, .2, .6 );
```

```
T4.Edit(.1, -1, .6 );
```

```
IEDer->DivideQuad(T1, T2, T3, T4, 9, 0, 0, 0, 1, 1, 1);
```

```
T1.Edit( .5, .2, -.2 );
```

```
T2.Edit(-.1, .2, 0 );
```

```
T3.Edit( .1, .2, .6 );
```

```
T4.Edit( .7, .2, .4 );
```

```
IEDer->DivideQuad(T1, T2, T3, T4, 7, 0, 0, 0, 1, 1, 1);
```

```

T1.Edit( .5, -1, -.2 );
T2.Edit( .5, .2, -.2 );
T3.Edit( .7, .2, .4 );
T4.Edit( .7, -1, .4 );
IEDer->DivideQuad(T1, T2, T3, T4, 8, 0, 0, 0, 1, 1, 1);
T1.Edit( .1, -1, .6 );
T2.Edit( .1, .2, .6 );
T3.Edit(-.1, .2, 0 );
T4.Edit(-.1, -1, 0 );
IEDer->DivideQuad(T1, T2, T3, T4, 10, 0, 0, 0, 1, 1, 1);
T1.Edit(-.1, -1, 0 );
T2.Edit(-.1, .2, 0 );
T3.Edit( .5, .2, -.2 );
T4.Edit( .5, -1, -.2 );
IEDer->DivideQuad(T1, T2, T3, T4, 11, 0, 0, 0, 1, 1, 1);
T1.Edit(-.15, -1, -.05);
T2.Edit(-.15, -.4, -.05);
T3.Edit( -.7, -.4, -.2);
T4.Edit( -.7, -1, -.2);
IEDer->DivideQuad(T1, T2, T3, T4, 16, 0, 0, 0, 1, 1, 1);
T1.Edit( -.7, -1, -.2);
T2.Edit( -.7, -.4, -.2);
T3.Edit( -.5, -.4, -.75);
T4.Edit( -.5, -1, -.75);
IEDer->DivideQuad(T1, T2, T3, T4, 15, 0, 0, 0, 1, 1, 1);
T1.Edit( .05, -1, -.6);
T2.Edit( .05, -.4, -.6);
T3.Edit(-.15, -.4, -.05);
T4.Edit(-.15, -1, -.05);
IEDer->DivideQuad(T1, T2, T3, T4, 13, 0, 0, 0, 1, 1, 1);
T1.Edit( -.5, -.4, -.75 );
T2.Edit( -.7, -.4, -.2 );

```

```

T3.Edit( -.15, -.4, -.05 );
T4.Edit( .05, -.4, -.6 );
IEDer->DivideQuad(T1, T2, T3, T4, 12, 0, 0, 0, 1, 1, 1);
T1.Edit( -.5, -1, -.75 );
T2.Edit( -.5, -.4, -.75 );
T3.Edit( .05, -.4, -.6 );
T4.Edit( .05, -1, -.6 );
IEDer->DivideQuad(T1, T2, T3, T4, 14, 0, 0, 0, 1, 1, 1);
__fastcall IEDerleyici::~~IEDerleyici()
{
    delete[] Patches;
}
void __fastcall IEDerleyici::Execute()
{
    while( !Terminated ){
        UpdateRadiance();
        iterations++;
    }
}
void IEDerleyici::SetMaxPtch( int a )
{
    if( a != MaxPtch ){
        if( Patches )
            delete[] Patches;
        Patches = new Triangle*[ a ];
        MaxPtch = a;
    }
}
void IEDerleyici::Reset()
{
    for( int a=PatchCounter-1; a>=0; a++){
        Patches[a]->IntensityAccumulatorR =

```

```

    Patches[a]->IntensityAccumulatorG =
    Patches[a]->IntensityAccumulatorB =
    Patches[a]->UnshotRadianceAccumulatorR =
    Patches[a]->UnshotRadianceAccumulatorG =
    Patches[a]->UnshotRadianceAccumulatorB = 0;
}
}

void IEDerleyici::QuadToTris(JVector VA, JVector VB, JVector VC, JVector VD,
int id, float ER, float EG, float EB, float Ror, float Rog, float Rob, float MyArea)
{
    if( PatchCounter < MaxPtch ){
        Patches[PatchCounter] = new Triangle;
        Patches[PatchCounter]->V1 = VA;
        Patches[PatchCounter]->V2 = VB;
        Patches[PatchCounter]->V3 = VC;
        Patches[PatchCounter]->EditNormal();
        Patches[PatchCounter]->plane_id = id;
        Patches[PatchCounter]->IntensityAccumulatorR = ER;
        Patches[PatchCounter]->IntensityAccumulatorG = EG;
        Patches[PatchCounter]->IntensityAccumulatorB = EB;
        Patches[PatchCounter]->UnshotRadianceAccumulatorR = ER;
        Patches[PatchCounter]->UnshotRadianceAccumulatorG = EG;
        Patches[PatchCounter]->UnshotRadianceAccumulatorB = EB;
        Patches[PatchCounter]->RoR = Ror;
        Patches[PatchCounter]->RoG = Rog;
        Patches[PatchCounter]->RoB = Rob;
        Patches[PatchCounter]->Area = MyArea;
        Patches[PatchCounter]->SetMaxPtch( MaxPtch );
        Patches[PatchCounter]->FormFactors[PatchCounter] = 0;
        PatchCounter++;
    }
}

```

```

if( PatchCounter < MaxPtch ){
    Patches[PatchCounter] = new Triangle;
    Patches[PatchCounter]->V1 = VC;
    Patches[PatchCounter]->V2 = VD;
    Patches[PatchCounter]->V3 = VA;
    Patches[PatchCounter]->EditNormal();
    Patches[PatchCounter]->plane_id = id;
    Patches[PatchCounter]->IntensityAccumulatorR = ER;
    Patches[PatchCounter]->IntensityAccumulatorG = EG;
    Patches[PatchCounter]->IntensityAccumulatorB = EB;
    Patches[PatchCounter]->UnshotRadianceAccumulatorR = ER;
    Patches[PatchCounter]->UnshotRadianceAccumulatorG = EG;
    Patches[PatchCounter]->UnshotRadianceAccumulatorB = EB;
    Patches[PatchCounter]->RoR = Ror;
    Patches[PatchCounter]->RoG = Rog;
    Patches[PatchCounter]->RoB = Rob;
    Patches[PatchCounter]->Area = MyArea;
    Patches[PatchCounter]->SetMaxPtch( MaxPtch );
    Patches[PatchCounter]->FormFactors[PatchCounter] = 0;
    PatchCounter++;
}
}

JVector JX = VB - VA;
JVector JY = VC - VB;
float MyArea = JX.Length()*JY.Length() / (DIVISOR*DIVISOR*2.0);
JVector Xdif = JX / DIVISOR;
JVector Ydif = JY / DIVISOR;
for(int i = 0; i < DIVISOR; i++)
{
    for(int j = 0; j < DIVISOR; j++)
    {
        JVector T1 = VA + Xdif*j;

```

```

        JVector T2 = T1 + Xdif;
        JVector T3 = T2 + Ydif;
        JVector T4 = T1 + Ydif;
        QuadToTris( T1, T2, T3, T4, id, ER, EG, EB, RoR, RoG,
RoB, MyArea );
    }
    VA = VA + Ydif;
}
}

void IEDerleyici::CalculateFormFactors(int i, int j)
{
    Triangle *I = Patches[i];
    Triangle *J = Patches[j];
    I->FormFactors[j] = 0;           // set initially to 0
    J->FormFactors[i] = 0;           // set initially to 0
    JVector Xi = ( I->V1 + I->V2 + I->V3 )/3 ; // Xi is near the middle of the
triangle i
    JVector Xj = ( J->V1 + J->V2 + J->V3 )/3 ;
    JVector Rij = Xj - Xi;
    float r = Rij.Length();
    Rij = Rij / r;
    JVector Rji = Rij * -1;
    float CosThetaI = Rij.Dot(I->Normal);
    float CosThetaJ = Rji.Dot(J->Normal);
    if( CosThetaI < 0 || CosThetaJ < 0 )
        return;
    for(int count = 0; count < PatchCounter; count++)
    {
        if( count != i && count != j
            && Patches[count]->plane_id != J->plane_id

            && Patches[count]->plane_id != I->plane_id )

```

```

        if(Patches[count]->Trace(Xi, Rij, r, 1) < r - threshold2)
            return;
    }
}

float DeltaF = CosThetaI * CosThetaJ / ( PI * r * r + J->Area );

if(DeltaF > 0){
    I->FormFactors[j] = DeltaF * J->Area;
    J->FormFactors[i] = DeltaF * I->Area;
}

}

void IEDerleyici::UpdateRadiance()
{
    float test = 0;
    float largest = 0;
    int i = 0;
    int count = 0;
    {
        test = Patches[count]->Area * ( Patches[count]-
>UnshotRadianceAccumulatorR +
        Patches[count]->UnshotRadianceAccumulatorB );

        if(test > largest)
        {
            largest = test;
            i = count;
        }
    }
    for(int j = 0; j < PatchCounter; j++)
    {
        if( Patches[j]->FormFactors[i] < 0 ) // If Fji is unknown

```

```

        CalculateFormFactors(j, i);
    }
}

//
Find it - this is the "progressive" part as form factors are
    if(Patches[j]->FormFactors[i] > 0)
    {
        float DeltaRadR = Patches[i]->UnshotRadianceAccumulatorR * Patches[j]-
        >RoR * Patches[j]->FormFactors[i];
        float DeltaRadG = Patches[i]->UnshotRadianceAccumulatorG
        * Patches[j]->RoG * Patches[j]->FormFactors[i];
        float DeltaRadB = Patches[i]->UnshotRadianceAccumulatorB
        * Patches[j]->RoB * Patches[j]->FormFactors[i];

        Patches[j]->UnshotRadianceAccumulatorR += DeltaRadR;
        Patches[j]->UnshotRadianceAccumulatorG += DeltaRadG;
        Patches[j]->UnshotRadianceAccumulatorB += DeltaRadB;
        Patches[j]->IntensityAccumulatorR += DeltaRadR;
        Patches[j]->IntensityAccumulatorG += DeltaRadG;
        Patches[j]->IntensityAccumulatorB += DeltaRadB;
    }
}

Patches[i]->UnshotRadianceAccumulatorR = 0;
Patches[i]->UnshotRadianceAccumulatorG = 0;
Patches[i]->UnshotRadianceAccumulatorB = 0;

#ifdef TRIANGLE_H
#define TRIANGLE_H
#include "JVector.h"
class Triangle{
public:
    Triangle();
    ~Triangle();

```

```

float Trace(JVector Start, JVector Direction, float current_t, int Depth) const;
void EditNormal();
void UpdateArea();
void SetMaxPtch( int );
JVector V1, V2, V3;
JVector Normal;
float D;
int plane_id;
float IntensityAccumulatorR;
float IntensityAccumulatorG;
float IntensityAccumulatorB;
float UnshotRadianceAccumulatorR;
float UnshotRadianceAccumulatorG;
float UnshotRadianceAccumulatorB;
float RoR;
float RoG;
float RoB;
float Area;
float *FormFactors;
};
#endif

```

//IŞIN İZLEME KODLARI

```
/* intensity.cpp */
```

```
#include "intensity.h"
```

```
Intensity::Intensity(Real red, Real green, Real blue) {
```

```
    iR = red;
```

```
    iG = green;
```

```
    iB = blue;
```

```
    r = (int) (iR * 255.0);
```

```
    g = (int) (iG * 255.0);
```

```
    b = (int) (iB * 255.0);
```

```

        if(r > 255) r = 255;
        if(g > 255) g = 255;
        if(b > 255) b = 255;
    }
    Intensity::Intensity() {
        iR = 0.0;
        iG = 0.0;
        iB = 0.0;
        r = (int) (iR * 255.0);
        g = (int) (iG * 255.0);
        b = (int) (iB * 255.0);
        if(r > 255) r = 255;
        if(g > 255) g = 255;
        if(b > 255) b = 255;
    }
    Intensity operator + (const Intensity& i1, const Intensity& i2) {
        return Intensity(i1.iR + i2.iR, i1.iG + i2.iG, i1.iB + i2.iB);
    }
    Intensity operator - (const Intensity& i1, const Intensity& i2) {
        return Intensity(i1.iR - i2.iR, i1.iG - i2.iG, i1.iB - i2.iB);
    }
    Intensity operator - (const Intensity& i1)
    {return Intensity(- i1.iR, - i1.iG, - i1.iB);
    }
    Intensity operator * (const Intensity& i1, const Intensity& i2) {
        return Intensity(i1.iR * i2.iR, i1.iG * i2.iG, i1.iB * i2.iB);
    }
    Intensity operator / (const Intensity& i1, const Intensity& i2) {
        return Intensity(i1.iR / i2.iR, i1.iG / i2.iG, i1.iB / i2.iB);
    }
    Intensity operator / (const Intensity& i, Real k) {
        return Intensity(i.iR / k, i.iG / k, i.iB / k);
    }

```

```

}
Intensity operator * (const Intensity& i, Real k) {
    return Intensity(i.iR * k, i.iG * k, i.iB * k);
}
Intensity operator * (Real k, const Intensity& i) {
    return i * k;
}
Intensity exp (const Intensity& i)SERKAN
{
    return Intensity(exp(i.iR), exp(i.iG), exp(i.iB));
}
#ifdef _PICTURE_H
#define _PICTURE_H

#include "intensity.h"
#include "format.h"
template<class EXTENSIONL>
class Picture {
public:
    Picture() : data(NULL) {}
    virtual ~Picture() { delete[] data; }
    int getWidth() { return p.width; }
    int getHeight() { return p.height; }
    int getDepth() { return p.depth; }
    void * getData() { return data; }
    int load(const char *file) {
        delete[] data;

        p = T_EXTENSIONL.load(file);
        if(p.data == NULL) return 1;

        if(p.depth == 8) {

```

```

        data = new Real[p.width * p.height];
        Real *ptr = (Real *) data;
        for(int i = 0 ; i < p.width * p.height ; i++) {
            ptr[i] = p.data[i] / 255.0;
        }

        } else if(p.depth == 24) {
            data = new Intensity[p.width * p.height];
            Intensity *ptr = (Intensity *) data;
            for(int i = 0 ; i < p.width * p.height ; i++) {
                in intensity.cpp
                    ptr[i] =
Intensity(p.data[i*3+2] / 255.0,
255.0,
255.0);
                    p.data[i*3+1] /
                    p.data[i*3] /
                }
            }
            delete [] p.data;
            return 0;
        }

static int save(const char *file, Intensity *buffer,
                int width, int height) {
    unsigned char *ptr;
    PictureDescriptor pict;
    pict.width = width;
    pict.height = height;
    pict.depth = 24;

    pict.data = new unsigned char[pict.width * pict.height * 3];
    ptr = pict.data;
    for(int i = 0 ; i < pict.width * pict.height ; i++, ptr+=3) {

```

```

        buffer[i].pack24(ptr);
    }
    return EXTENSIONL::save(file, pict);
    delete[] pict.data;
}

static int save(const char *file, int *buffer,
               int width, int height) {
    unsigned char *ptr;
    PictureDescriptor pict;
    pict.width = width;
    pict.height = height;
    pict.depth = 24;
    * pict.height * 3];
    ptr = pict.data;
    for(int i = 0 ; i < pict.width * pict.height ; i++, ptr+=3) {
        *(ptr+2) = (buffer[i] & 0xFF0000) >> 16;
        *(ptr+1) = (buffer[i] & 0xFF00) >> 8;
        *(ptr) = (buffer[i] & 0xFF);
    }
    return EXTENSIONL::save(file, pict);
    delete[] pict.data;
}

private:
    PictureDescriptor p;
    void *data;
    EXTENSIONL T_EXTENSIONL;
};

#endif
#include <vcl.h>
#pragma hdrstop
#include <LocUtils.hpp>

```

```

USERES("Gonio.res");
USEFORM("AnaPen.cpp", AnaPen);
USELIB("PortSurucu\PSLib.lib");
USEFORM("YeniAd.cpp", YeniAd);
USELIB("IEDerleyici\IELib.lib");
USELIB("IIDerleyici\IILib.lib");
USELIB("IsikModelleyici\IMLib.lib");
USELIB("IsikSiddetEgrisi\SELib.lib");
USELIB("IsikKaynagiModeli\IKLib.lib");
USELIB("IsikKaynagiModeli\_lib95841.lib");
USELIB("IsikKaynagiModeli\_lib95842.lib");
USELIB("IsikKaynagiModeli\_lib95843.lib");
USELIB("IsikKaynagiModeli\_lib95844.lib");
USELIB("IsikKaynagiModeli\_lib95845.lib");
USELIB("Raytressi\rytrcr.lib");
USEFORM("CvrPen.cpp", CevirPen);
USERES("Gonio_klr.res");
USEUNIT("C:\Program Files\KLib\localizer\lib\LocUtils.pas");
USERC("PrjConst.rc");
USERC("PortSurucu\PrjConst.rc");
USERC("IsikModelleyici\PrjConst.rc");
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Localizer->InitReg("Software\\Gonio");
        Application->Initialize();
        Application->CreateForm(__classid(TAnaPen), &AnaPen);
        Application->CreateForm(__classid(TYeniAd), &YeniAd);
        Application->CreateForm(__classid(TCevirPen), &CevirPen);
        Application->Run();
    }
}

```

```
catch (Exception &exception)
{
    Application->ShowException(&exception);
}
return 0;
}
//-----serkan
```



ÖZGEÇMİŞ

11.05.1974 yılında Isparta'da doğdu. Liseyi, Isparta Teknik Lisesi Elektrik Bölümünde 1992 yılında tamamlayarak aynı yıl Gazi Üniversitesi Teknik Eğitim Fakültesi Elektrik Bölümüne girdi. 1996 yılının Güz döneminde Gazi Üniversitesi Fen Bilimleri Enstitüsü Elektrik Eğitimi Anabilim Dalında yüksek lisans çalışmalarına başladı. Ocak 1997' de Gazi Üniversitesi Elektrik Eğitimi Bölümü Aydınlatma ve Tesisat Eğitimi Anabilim Dalında araştırma görevlisi görevine atandı. Ocak 1999 yılında Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Eğitimi Anabilim Dalında, "Işık şiddet eğrilerinin bilgisayar yardımıyla çıkartılması" konulu bir tez hazırlayarak yüksek lisans derecesi aldı. 1999 yılı Şubat ayında doktora çalışmalarına başladı. 2002 yılının Haziran ayında öğretim görevlisi oldu. Aynı bölümde çalışmaya devam etmektedir.