KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

TERSİNE MÜHENDİSLİK UYGULAMALARINDA NOKTA BULUTU VERİLERİNDEN ÖRME YÜZEY VE PARAMETRİK YÜZEY DENKLEMLERİNİN ELDE EDİLMESİ

Cengiz BALTA

KOCAELİ 2016

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ

ANABİLİM DALI

DOKTORA TEZİ

TERSİNE MÜHENDİSLİK UYGULAMALARINDA NOKTA BULUTU VERİLERİNDEN ÖRME YÜZEY VE PARAMETRİK YÜZEY DENKLEMLERİNİN ELDE EDİLMESİ

CENGİZ BALTA

Yrd. Doç. Dr. Sıtkı ÖZTÜRK		
Danışman,	**Kocaeli Üniversitesi**	.....
Prof. Dr. Zafer BİNGÜL		
Jüri Üyesi,	**Kocaeli Üniversitesi**	.....
Doç. Dr. Mustafa ÇAKIR		
Jüri Üyesi,	**Kocaeli Üniversitesi**	.....
Prof. Dr. Cemil ÖZ		
Jüri Üyesi,	**Sakarya Üniversitesi**	.....
Yrd. Doç. Dr. Hayrettin EVİRGEN		
Jüri Üyesi,	**İstanbul Üniversitesi**	.....

Tezin Savunulduğu Tarih: 04.11.2016

ÖNSÖZ VE TEŞEKKÜR

Bu tez, geometri, bilgisayar grafikleri ve bilgisayar destekli tasarım alanlarından çeşitli parçalar içermektedir. Ülkemizde bilgisayarlı geometri temellerinin oluşması ve yaşayan matematik uygulamalarının gelişmesi açısından bir katkı olmasını ümit ederim.

Doktora eğitimi boyunca sabır gösteren tüm arkadaşlarıma ve danışman hocama teşekkür ederim.

Kasım – 2016 Cengiz BALTA

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR i
İÇİNDEKİLER ii
ŞEKİLLER DİZİNİ iv
TABLOLAR DİZİNİ vii
SİMGELER VE KISATMALAR DİZİNİ viii
ÖZET ix
ABSTRACT x
GİRİŞ 1
1. TERSİNE MÜHENDİSLİK UYGULAMALARI 3
1.1. Tersine Mühendislikte Aşamalar 3
1.2. Veri Toplama Metotları 4
2. NOKTA BULUTUNDAN ÖRME YÜZEY ELDE ETME METOTLARI 8
2.1. Yüzey Geri Çatma Probleminin Tanımı 8
2.2. Uzayda Hiyerarşik Bölümleme 10
2.2.1. Bsp-tree 10
2.2.2. kD-tree 10
2.2.3. Quadtree/octree 11
2.2.4. Çevreleyen hacim yapısı, bounding volume 13
2.3. Sınırlayan Küpler Algoritması - Marching Cubes 13
2.4. Space Carving – Boşluk Yontma 14
2.5. Örme Yüzey Üretme Algoritmaları 15
2.6. Delaunay Temelli Algoritmalar 17
2.7. Yerel Bazlı Algoritmalar 18
2.7.1. k-en yakın komşuların bulunması 19
2.7.2. Temel bileşenler analizi (principle component analysis, pca) 19
2.7.3. Yerel – grid bazlı algoritmalar 21
2.7.3.1. Tanjant düzlem yaklaşımı 21
2.7.3.2. Top yuvarlama - ball pivoting algoritması 23
2.7.4. Kapalı fonksiyon yüzey algoritmaları 23
2.7.4.1. Moving least squares 24
2.7.4.2. Radyal tabanlı fonksiyonlar 25
2.8. Metotların Karşılaştırılması 25
3. ÖRME YÜZEY SADELEŞTİRME ALGORİTMALARI 27
3.1. Yerel Sadeleştirme Algoritmaları 28
3.1.1. Köşe yok etme (vertex decimation) 29
3.1.1.1. Köşe sınıflandırma (vertex classification) 29
3.1.1.2. Yok etme kriteri (decimation criterion) 30
3.1.1.3. Üçgenleme 30
3.1.2. Kenar kaynaştırma (Edge Contraction) 31
3.2. Global Sadeleştirme Algoritmaları 32
3.2.1. Poligon köşelerinin birleştirilmesi (vertex clustering) 32
3.2.2. Örnekleme (sampling) 32
3.3. Dörtgensel Bölgelere Ayırma 32

4. B-SPLİNE EĞRİ VE YÜZEYLER 34
4.1. B-Spline Düğüm Vektör Yaklaşımı 38
4.2. B-Spline Baz Fonksiyonlarının Özyinelemeli Hesaplanması 40
4.3. B-spline Eğri Anlık Değer Hesabı 44
4.4. B-Spline Yüzey Tanımı 45
4.5. B-spline Eğri Enterpolasyonu 48
4.6. B-spline Yüzey Enterpolasyonu 51
4.7. Veri Parametrelendirme Yöntemleri 53
4.7.1. Düzgün dağılımlı parametrelendirme - uniform spaced 53
4.7.2. Kiriş uzunluğuna göre parametrelendirme - chord length 56
4.7.3. Merkezcil parametrelendirme - centripetal method 57
4.7.4. Düğüm vektörü üretimi 59
4.7.5. Evrensel metot - universal method 61
4.7.6. Foley-Nielson metodu 62
4.7.7. Önerilen metot: dinamik merkezcil metot 64
4.7.8. Eğri kıyaslamaları 66
4.8. Polinom Derecelerine Göre Parametrelendirme Başarımı 77
4.9. B-Spline Eğrilerin Türevleri 82
4.10. Yüzey İçin Parametrelendirme ve Düğüm Vektörü Üretimi 84
4.11. Mesh Yüzeyden Grid Data Üretimi ve Yüzey Enterpolasyonu 86
4.12. 3D Hata Analizleri 89
5. SONUÇLAR VE ÖNERİLER 97
KAYNAKLAR 100
KİŞİSEL YAYIN VE ESERLER 105
ÖZGEÇMİŞ 106

ŞEKİLLER DİZİNİ

Şekil 1.1. Tersine mühendislik aşamaları 4
Şekil 1.2. Veri toplama metotları 5
Şekil 1.3. Acuity AR700 laser mesafe sensörü 5
Şekil 1.4. Faro laser tarama robot kolu 6
Şekil 1.5. CMM temaslı ölçme 6
Şekil 1.6. Brontes diş tarama 6
Şekil 2.1. Nokta bulutu ve elde edilen örme yüzey 8
Şekil 2.2. STL formatı, StereoLithography Interface Specification 9
Şekil 2.3. BSP-Tree hiyerarşik ağaç yapısı 10
Şekil 2.4. kD-Tree oluşturma 11
Şekil 2.5. kD-Tree yapısına yerleştirilmesi için gerekli kod yapısı 11
Şekil 2.6. QuadTree: dörtlü ağaç ile uzay bölümleme 12
Şekil 2.7. Octree: Sekizli ağaç yapısı ile bölümleme 12
Şekil 2.8. Öz yinelemeli sekizli ağaç yapısının basit bir kod yapısı 12
Şekil 2.9. Çevreleyen küreler yapısı ile yüzey geri çatma uygulaması 13
Şekil 2.10. Sınırlayan kareler 13
Şekil 2.11. Sınırlayan kareler için kombinasyonlar 14
Şekil 2.12. Sınırlayan küpler için kombinasyonlar 14
Şekil 2.13. Sınırlayan küpler, 150 dilimli MR verisi üzerinden çıkarım 14
Şekil 2.14. Boşluk yontma algoritmasının aşamalı gösterimi 15
Şekil 2.15. İki boyutlu gösterim 15
Şekil 2.16. Delaunay üçgenleme 17
Şekil 2.17. Delaunay üçgenleme ve karşılık voronoi diyagramı 18
Şekil 2.18. Crust algoritması 18
Şekil 2.19. Cocone algoritması 19
Şekil 2.20. BDD tree yapısı 20
Şekil 2.21. P noktası ve yakın komşuları için temel bileşenler analizi 20
Şekil 2.22. Hoppe mesh üretme algoritması 21
Şekil 2.23. VTK kütüphanesi ile nokta bulutu gösterimi 22
Şekil 2.24. Nokta bulutundan elde edilen örme yüzey 22
Şekil 2.25. Ball pivoting algoritması 23
Şekil 2.26. Ohtake, nokta bulutu işleme aşamaları 25
Şekil 3.1. Köşe Sınıflandırma 29
Şekil 3.2. Sınırlayan köşe ve dahili köşe hata ölçütü 30
Şekil 3.3. VTK kütüphanesi ile yapılan mesh sadeleştirme işlemi 31
Şekil 3.4. Kenar kaynaştırma işlemi 32
Şekil 3.5. Kademeli sadeleştirilen model 32
Şekil 3.6. RapidForm b-spline yama ekran görüntüsü 33
Şekil 4.1. Pierre Bezier, fikirlerini açıklıyor 34
Şekil 4.2. Rich Riesenfeld, 1970 34
Şekil 4.3. Boeing teknik ressam, fiziksel spline çizimi 35
Şekil 4.4. İkinci derece b-spline gösterimi 36
Şekil 4.5. Kübik uniform b-spline yapılanması 37

Şekil 4.6. Çan eğrisi, dört parçalı olarak tanımlanıyor 39
Şekil 4.7. Ağırlık fonksiyonları, dört parçalı çan eğrilerinden oluşur 39
Şekil 4.8. 0-1 Aralığında ağırlık fonksiyonları 39
Şekil 4.9. p'inci derece baz fonksiyonları tablosu 41
Şekil 4.10. Sıfır derece baz fonksiyonları 41
Şekil 4.11. Birinci derece baz fonksiyonları 42
Şekil 4.12. İkinci derece baz fonksiyonları 42
Şekil 4.13. Parametre uzayı ve nesne uzayının bağımsızlığı 42
Şekil 4.14. 2. derece baz fonksiyonlar 44
Şekil 4.15. (a). İkinci derece baz fonksiyonları (b). b-spline eğri 46
Şekil 4.16. Bi-kübik b-spline yüzey segmenti 47
Şekil 4.17. Kartezyen çarpım gösterimi 48
Şekil 4.18. Parametreler ve karşılık gelen veri noktaları 54
Şekil 4.19. Çeşitli parametrelendirmelere göre üretilen eğriler 54
Şekil 4.20. Parametrelendirme sırası değişiminin enterpolasyona etkisi 54
Şekil 4.21. Düzgün dağılımlı parametrelendirme 55
Şekil 4.22. Uniform (kesik çizgili) ve chord length parametrelendirme 57
Şekil 4.23. Chord length, son segmentte hata 57
Şekil 4.24. Parametrelendirme metotları 59
Şekil 4.25. Parametrelendirme metotları 59
Şekil 4.26. Parametre ve düğüm değerleri 61
Şekil 4.27. Seçilen parametreler ve ortalama ile düğüm vektörleri 61
Şekil 4.28. Baz fonksiyon tepeleri – parametreler 62
Şekil 4.29. Foley metodunda kullanılan parametreler 64
Şekil 4.30. Birinci data set ile kübik eğri enterpolasyonu 67
Şekil 4.31. Birinci data set için e_i üs değerleri 68
Şekil 4.32. Birinci data set için metotlara göre eğriler 69
Şekil 4.33. İkinci data set ile kübik eğri enterpolasyonu 70
Şekil 4.34. İkinci data set için ei üs değerleri 71
Şekil 4.35. İkinci data set için metotlara göre eğriler 71
Şekil 4.36. Üçüncü data set ile kübik eğri enterpolasyonu 72
Şekil 4.37. Üçüncü data set için ei üs değerleri 73
Şekil 4.38. Üçüncü data set için metotlara göre eğriler 73
Şekil 4.39. Dördüncü data set ile kübik eğri enterpolasyonu 74
Şekil 4.40. Dördüncü data set için ei üs değerleri 75
Şekil 4.41. Dördüncü data set için metotlara göre eğriler 76
Şekil 4.42. Birinci data set ile eğri enterpolasyonu 77
Şekil 4.43. İkinci data set ile eğri enterpolasyonu 78
Şekil 4.44. Üçüncü data set ile eğri enterpolasyonu 80
Şekil 4.45. Dördüncü data set ile eğri enterpolasyonu 81
Şekil 4.46. Birinci data set için türevler 83
Şekil 4.47. İkinci data set için türevler 83
Şekil 4.48. Üçüncü data set için türevler 84
Şekil 4.49. Dördüncü data set için türevler 84
Şekil 4.50. u yönünde parametre hesabı 85
Şekil 4.51. Parametrelendirme izdüşüm yönü, 1-1 örten (bijective) 86
Şekil 4.52. Matlab düzensiz veri, lineer enterpolasyon 87
Şekil 4.53. Mesh parametrelendirme neticesi, Peyre/Floater 87
Şekil 4.54. Mesh yüzey ve elde edilen grid data 87

Şekil 4.55. Grid data noktalarının ayrık gösterimi 88
Şekil 4.56. Grid data – b-spline geçiş aşamaları 88
Şekil 4.57. B-spline yüzey..... 89
Şekil 4.58. IGES formatlı b-spline yüzey, pro engineer gösterimi 89
Şekil 4.59. 8x8 grid data..... 90
Şekil 4.60. 70x70 lineer dağıtılmış grid data..... 90
Şekil 4.61. Merkezcil metoda göre üretilen eğri ve hata vektörleri 91
Şekil 4.62. Merkezcil metoda göre hata 91
Şekil 4.63. Önerilen metoda göre üretilen eğri ve hata vektörleri..... 92
Şekil 4.64. Önerilen metoda göre hata 92
Şekil 4.65. Önerilen metod, e_i dinamik üs değerleri..... 92
Şekil 4.66. Önerilen metod, e_i üs değerleri..... 93
Şekil 4.67. 70x70 lineer dağıtılmış grid data, ikinci data set 93
Şekil 4.68. Centripetal metoda göre üretilen eğri ve hata vektörleri..... 94
Şekil 4.69. Centripetal metod, ikinci data set için hata 94
Şekil 4.70. Önerilen metoda göre üretilen eğri ve hata vektörleri..... 95
Şekil 4.71. Önerilen metod, ikinci data set için hata, toplam..... 95
Şekil 4.72. Önerilen metod, e_i dinamik üs değerleri 96
Şekil 4.73. Önerilen metod, e_i üs değerleri..... 96

TABLOLAR DİZİNİ

Tablo 2.1. Metotların iki boyutlu gösterimi 25
Tablo 2.2. Metotların karşılaştırılması 26
Tablo 4.1. Eğri segmenti boyunca kirişten sapmalar, şekil 4.30..... 69
Tablo 4.2. Eğri segmenti boyunca kirişten sapmalar, şekil 4.33..... 71
Tablo 4.3. Eğri segmenti boyunca kirişten sapmalar, şekil 4.36..... 74
Tablo 4.4. Eğri segmenti boyunca kirişten sapmalar, şekil 4.39..... 76

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

3D	: 3 Dimensions (Üç Boyutlu)
BDD	: Binary Decision Tree (İkili Karar Ağacı)
B-SPLINE	: Basis Splines (B-spline)
CAD	: Computer Aided Design (Bilgisayar Destekli Tasarım)
CAGD	: Computer Aided Geometric Design (Bilgisayar Destekli Geometrik Tasarım)
CAM	: Computer Aided Manufacturing (Bilgisayar Destekli İmalat)
CMM	: Coordinate Measuring Machine (Koordinat Ölçme Makinası)
CNC	: Computer Numeric Controller (Bilgisayarlı Sayısal Kontrol)
CT	: Computer Tomography (Bilgisayarlı Tomografi)
IGES	: Initial Graphics Exchange Specification (İlk Grafik Değişim Standartı)
LOD	: Level of Details (Kademeli Detay Seviyeleri)
MR	: Manyetik Rezonans
PCA	: Principle Components Analysis (Temel Bileşenler Analizi)
RBF	: Radial Basis Functions (Radial Tabanlı Fonksiyonlar)
STL	: Standard Tessellation Language, Stereo Lithography (Standart Mozikleme Dili)
VTK	: Visualization Toolkit (Görsel Araç Seti)

TERSİNE MÜHENDİSLİK UYGULAMALARINDA NOKTA BULUTU VERİLERİNDEN ÖRME YÜZEY VE PARAMETRİK YÜZEY DENKLEMLERİNİN ELDE EDİLMESİ

ÖZET

Bu çalışmada, çeşitli veri toplama yöntemleri ile dış dünyadan elde edilen, nokta bulutu olarak ifade edilen, cisim yüzey ve hacim bilgisini ifade eden ham verilerin, bilgisayar ortamında gösteriminde ilk aşama olan örme yüzey üretilmesi ele alınıyor. Tersine mühendislik uygulamalarında, tarama cihazları ile elde edilen nokta bulutu verilerinin, bilgisayarda gösterim seviyesinde örme yüzey ve bir üst seviyede b-spline yama şeklinde geri elde edilmesi inceleniyor.

B-spline eğri ve yüzey giydirmede ilk aşama olarak parametrelendirme yapılması ve düğüm vektörünün belirlenmesi gerekir. Bu amaçla, uniform, chord length, centripetal, Foley ve benzeri metotlar geliştirilmiştir.

Uniform metot, data noktalarının düzenli olması halinde iyi sonuç vermektedir. Chord-kiriş mesafesine göre parametrelendirme, uzun kirişlerde, istenmeyen salınımlar üretebilmektedir. Bundan dolayı kiriş mesafesinin karekköküne göre işleyen centripetal metot geliştirilmiştir. Bu modelde kirişlerin üs kuvveti olarak sabit bir sayı önerilmiştir. Bu tezde ise, sabit bir üs değeri yerine, kiriş mesafesine bağlı olarak değişen dinamik bir üs derecesi ile işleyen yeni bir parametrelendirme metodu önerilmiştir.

Anahtar Kelimeler: B-spline, Örme Yüzey, Parametrelendirme, Tersine Mühendislik.

OBTAINING MESH AND PARAMETRIC SURFACE EQUATIONS FROM POINT CLOUD DATA IN REVERSE ENGINEERING APPLICATIONS

ABSTRACT

In this work, we discuss the methods for 3D data collection, how to obtain mesh from scattered point cloud data and how to obtain b-spline patches from this mesh in reverse engineering applications.

B-spline data interpolation and approximation needs parameterization at first step. For this purpose many algorithms developed like uniform, centripetal, chord length, Foley and universal methods. Uniform method gives a good result in case of regular data points are given as an input set. Chord length method can produce large deflection of curve from chord for long chords. To be able to remove this effect centripetal method is developed. Centripetal method uses a fixed power of chord lengths for parameter distribution.

In this thesis we propose an improved version of centripetal parameterization method for b-spline data interpolation. While traditional centripetal method using fixed powers related with chord lengths of data points, this method calculates individual powers for each chord.

Keywords: B-spline, Mesh, Parameterization, Reverse Engineering.

GİRİŞ

Bu tez çalışmasında, laser tarama ve benzeri sensörler ile elde edilen ham nokta bulutu verilerinden, örme yüzey elde edilmesi ve yüzeylerin bölgelere ayrıştırılarak, CAD-CAM sistemlerinde kullanılan b-spline yüzey modelleme metotları ile parametrik olarak ifade edilebilir seviyeye kadar yükseltgenmesi amaçlanmıştır.

Tezin birinci bölümünde tersine mühendislik uygulamalarında kullanılan aşamalar ve veri toplama yöntemleri anlatılmaktadır. Tersine mühendislik uygulamalarında, veri toplama, boşlukları doldurma, yumuşatma, seyreltme gibi ön işlemler ve sonrasında üçgen mesh yüzey elde edilmesi gibi aşamalar bulunmaktadır. Veri toplama metotlarında ise, temaslı veya temassız, çeşitli sensörler yardımı ile, cisim yüzeyinin bilgisayar ortamında anlamlı ve işlenebilir veri haline getirilmesi için kullanılan yöntemler anlatılmaktadır.

İkinci bölümde ise, grafik verilerini işlemek üzere bilgisayar ortamında kullanılan veri yapıları temel seviyede incelenmiştir. Sonrasında, sensörler yardımı ile elde edilen nokta bulutu verilerinden, üçgen mesh – örme yüzeyler elde edilmesi için kullanılan mesh üretme metotları gösterilmiştir.

Nokta bulutu verilerinden üretilen örme yüzey, genellikle istenmeyen fazlalıklar içermektedir. Bundan dolayı mesh üzerinde çeşitli sadeleştirmeler yapılması gerekebilir. Bu amaçla, örme yüzey sadeleştirme algoritmaları üçüncü bölümde verilmiştir.

Dördüncü bölümde, otomotiv, gemi ve uçak sanayiinin matematiksel temelini hazırlayan, b-spline eğri ve yüzeylerin matematiksel altyapısı anlatılmış, bu matematiksel altyapının bilgisayarlı hesaplamasında kullanılan öz yinelemeli – recursive yapıları incelenmiştir. B-spline eğri anlatımından sonra, b-spline yüzeylerin 1artezyen çarpım olarak ifadesi gösterilmiştir. Sonrasında ise, tersine mühendislik uygulama amacına yönelik olarak, hedef noktalardan geçen eğri ve yüzey enterpolasyonu amacıyla, hedef data noktalarının parametrelendirilmesi, bu

parametreler kullanılarak, bilinmeyen eğri ve yüzeyi ifade edecek düğüm vektörünün üretilmesi ve eğri/yüzey enterpolasyon metotları gösterilmiştir.

Dördüncü bölümün devamında, mevcut eğri parametrelendirme metotlarına alternatif olarak kiriş mesafesine göre dinamik üs hesaplayarak işleyen bir parametrelendirme metodu önerilmiştir.

Bölümün sonunda, örnek bir mesh yüzeyden tek parçalı bir b-spline yüzey elde edilerek, bu yüzey IGES formatına dönüştürülerek, Pro Engineer programı ile gösterilmiştir. Bu aşama ile, CAM yazılımları kullanılarak CNC tezgahı ile mesh yüzeyin fiziksel olarak tekrar üretilebilir hale getirilmesi amaçlanmıştır.

1. TERSİNE MÜHENDİSLİK UYGULAMALARI

1.1. Tersine Mühendislikte Aşamalar

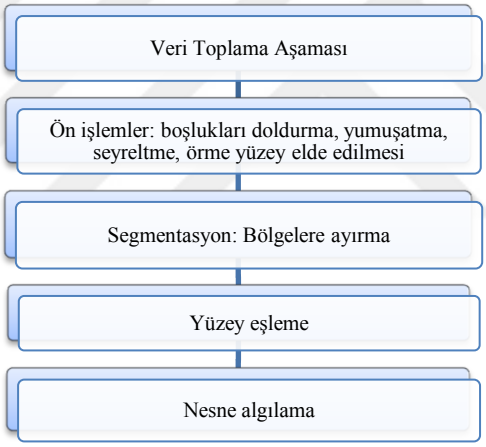
Nokta bulutu verileri, mekanik parçalar, kalıpçılık, tıbbi görüntüleme, haritacılık, arazi yüzeyi modelleme, bina ve yapı modelleme ve benzeri konularda, cisim ve nesneleri temsil eden ham veri olarak, tarama cihazlarından elde edilir. Ham veri, uzayda n adet (x,y,z) nokta bilgisi içerir. Bu ham veriden, bilgisayar ortamında verimli bir şekilde gösterimine uygun olan örme yüzeyin elde edilmesi için, çeşitli aşamalarda işleyen algoritmalar uygulanır. Tersine mühendislik uygulamaları, bu örme poligon yapısı veya nokta bulutu verisi üzerinde, çeşitli sınıflandırma algoritmaları kullanarak, ilişkili yüzey bölgelerini ayırıp, her bir bölgeye parametrik yüzeyler giydirerek, nokta bulutu verisinin daha üst seviyede parametrik olarak ifade edilmesini sağlar.

Laser tarama veya dokunmalı koordinat ölçüm sistemleri ile sahadan toplanan nokta bulutu verisinden, bilgisayar ortamında gösterime uygun yüzeylerin elde edilmesi, oldukça zor ve karmaşık işlemler gerektirir. Özellikle sahadan elde edilen verinin gürültülü ve düzensiz olması problemin çözümünü daha da karmaşık hale getirir.

Bilgisayar ortamında yüksek kalitede gösterim sağlamak üzere poligon modellerin üretilmesi gerekir. Video oyunları, animasyonlar, sanal gerçeklik ve simülasyon uygulamaları nesneleri poligon modeli olarak işlerler. Burada temel amaç, en düşük hesap gücü ile cisimlerin yüksek kalitede bilgisayar ortamında gösterimini sağlamaktır. Endüstriyel üretim ortamlarında ise, bilgisayar destekli tasarım ve imalat (CAD-CAM) yazılımları kullanılır. Bu yazılımlarda öncelik, cisimlerin ve parçaların ölçekli olarak tekrar üretilebilir parametrik modeller ile ifade edilmesidir. Tersine mühendislik uygulamalarında, tarama cihazından elde edilen nokta verilerinin poligon formatına dönüştürülmesi, parametrik yüzey eşleme ve nesne tanımlama aşamalarından bir önceki aşama olarak yer alır.

Tersine mühendislik ve uzaktan algılama ile görüntüleme uygulamaları, endüstrinin çeşitli dallarında uygulama alanı bulmuştur. Son yıllarda laser mesafe sensörlerinin gelişmesi ile birlikte, sahadan tutarlı ve yoğun veri toplamanın kolaylaşması ile birlikte, endüstriyel üretimler, CAD-CAM, arazi-bina tarama, bilgisayarlı görme-robotik konularında laser tarama sensörleri sıklıkla kullanılmaya başlamıştır. Manyetik rezonans (MR) ve bilgisayarlı tomografi (CT) metotları ile, cisimlerin dış yüzeylerinin yanı sıra, iç bölmelerinin de görüntülenmesi ve bilgisayar ortamına aktarılabilmesi imkanı oluşmuştur. Bu veri toplama teknikleri özellikle tıbbi görüntüleme alanında başarı ile uygulanmaktadır.

Tersine mühendislik uygulamalarının genel aşamaları şu şekilde özetlenebilir: Veri toplama, ön işlemler, bölümleme, yüzey eşleme, nesne algılama [1]. Şekil 1.1'de bu aşamalar gösterilmiştir.



Şekil 1.1. Tersine mühendislik aşamaları

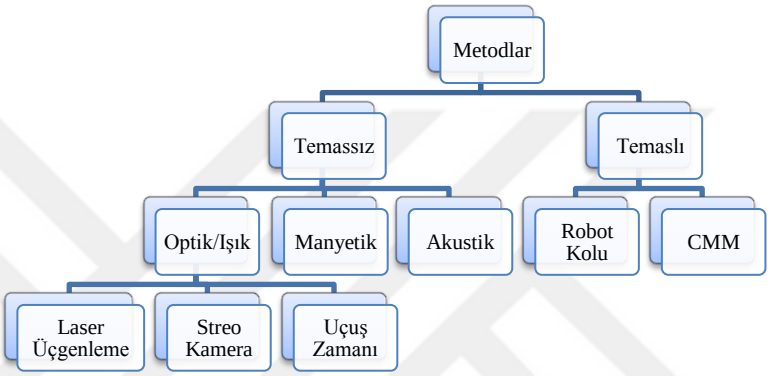
İlk iki aşama, bütün uygulama alanlarında, dış ortamdan elde edilen verinin bilgisayar ortamında, poligon formatında verimli gösterimini hedefler ve görselleştirme için yeterlidir. Diğer aşamalar, üretim ve analize yönelik CAD-CAM amaçlı uygulamalarda, yüzeyin parametrik eğri denklemleri ile ifade edilmesini hedefler.

1.2. Veri Toplama Metotları

Uzaktan algılamada, tersine mühendislik ve tıbbi görüntüleme amacı ile, dış ortamdan veri alınmasında çeşitli metotlar kullanılır. Bu metotlar, temaslı ve

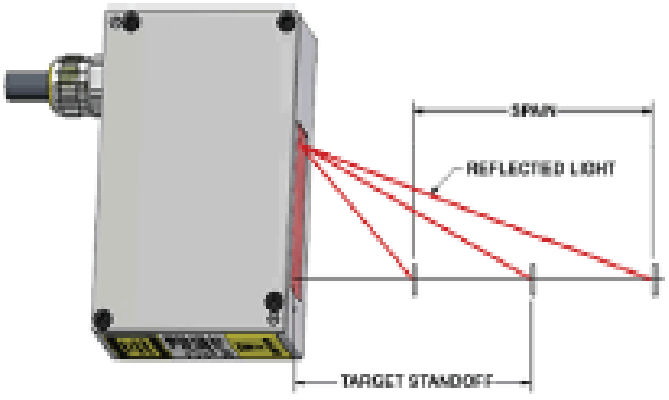
temassız olarak iki gruba ayrılabilir. Şekil 1.2'de bu metotların bir sınıflandırması gösterilmiştir.

Temaslı veri toplamada, robot kolu veya benzeri bir mekanizma, cisim üzerinde temas ederek dolaştırılarak koordinat bilgileri toplanır. Temassız metotlar, optik, manyetik ve akustik olarak sıralanabilir. Optik ve ışığın kullanıldığı metotlarda, streo kamera tekniği, uçuş zamanı ölçme tekniği ve üçgenleme – yansıma açısı ile mesafe ölçümü teknikleri genel başlıklar olarak sayılabilir [1].



Şekil 1.2. Veri toplama metotları

Şekil 1.3, 1.4, 1.5 ve 1.6'da endüstriyel olarak kullanılan çeşitli veri toplama sistemleri gösterilmiştir.



Şekil 1.3. Acuity AR700 laser mesafe sensörü

Veri toplama aşamasında karşılaşılan problemler şu şekilde sınıflandırılabilir:

- Kalibrasyon ve tutarlılık: Lens etkileri, lineer olmayan elektriksel davranışlar.
- Erişilebilirlik: Tarayıcı, gizli yüzeylere erişemeyebilir.
- Kapatılma/okolüsyon: Gölge veya engeller sebebi ile tarama bölgesi erişilemez.



Şekil 1.4. Faro laser tarama robot kolu



Şekil 1.5. CMM temaslı ölçme



Şekil 1.6. Brontes diş tarama

–Çoklu görüntüleme: Birden fazla tarama olması durumunda, verilerin aynı koordinat sisteminde kayıt edilmesi gerekir.

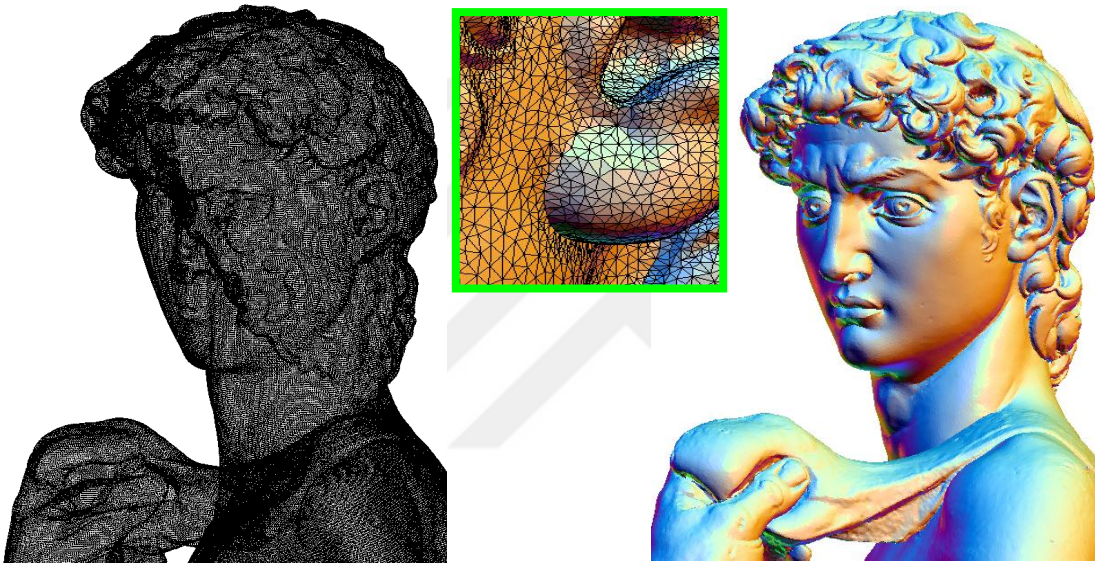
– Gürültü ve eksik veri: Hatalı veya eksik veri bölgelerinin düzeltilmesi gerekir.

–Parçaların dağılımları: Taramada alınan bir örnek parça, tüm parçaları temsil etmez. Yüzeyin tolerans dağılımı, geri çatmada önemli bir veridir [1].

Veri toplama aşamasında oluşan problemleri gidermek üzere, çeşitli ön işlemler uygulanır. Veri üzerinde, boşlukları doldurma, gürültü giderme, seyreltme ve benzeri düzeltmeler yapılır [2].

2. NOKTA BULUTUNDAN ÖRME YÜZEY ELDE ETME METOTLARI

Bu bölümde, algılama cihazları ile sahadan toplanan ham (x,y,z) nokta bulutu verisinden üçgen örme yüzey elde edilmesi için gerekli aşamalar anlatılıyor. Şekil 2.1'de Ohtake'nin çalışmadaki [22] nokta bulutu verisi girdisi ve örme yüzey çıktısı gösteriliyor.



Şekil 2.1. Nokta bulutu ve elde edilen örme yüzey [22]

2.1. Yüzey Geri Çatma Probleminin Tanımı

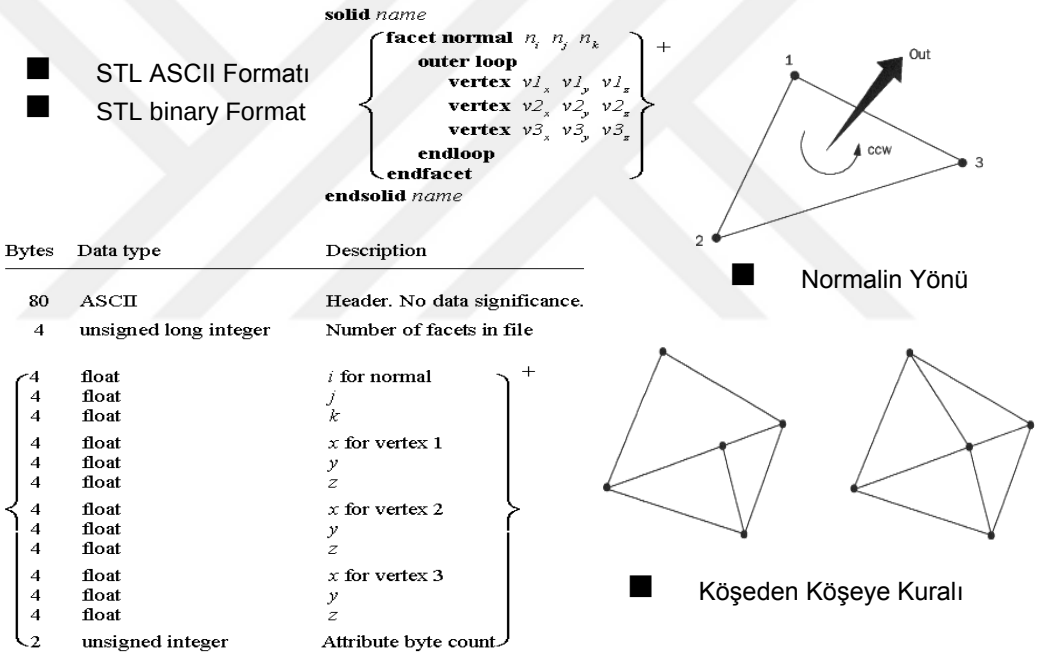
Yüzey geri çatma algoritmaları, tarama cihazlarından elde edilen yoğun verilerin, cismin geometrik ve topolojik özelliklerini koruyarak, daha verimli bir şekilde saklanmasını hedefler [3].

Bir kısım örme yüzey çatma algoritmaları, nokta kümesi ile verilen noktalara çeşitli fonksiyonlar ile yaklaşım yaptıklarından dolayı, üretilen yüzeyin noktaları, tam olarak orijinal yüzeyin noktaları ile örtüşmesi beklenmez.

Geri çatma algoritmaları, çeşitli dış değişkenlere bağımlıdır. Bazı algoritmalar, yoğun nokta bulutu verileri ile çalışır iken, bazıları tarayıcı cihazın satır taraması veya dilimlenmiş CT taraması yaptığını farz ederek 8artez – dilimli çevre çizgisi

bilgisini kullanır. Bir kısım tarama cihazları, yüzey normal bilgisini tarama zamanında ürettiğinden, bazı algoritmalar yüzey normalinin hazır olarak verildiğini farz eder. Bir kısım algoritmalarda ise, yüzey normali, noktaların komşuluk ilişkileri kullanılarak yaklaşımla bulunur. Bir kısım algoritmalar, gürültülü veya eksik verileri işleyebilir iken, çoğu algoritmalar, düzenli sıklıkta veya belirli bir sıklıkta örneklenmiş nokta verisi ile verimli olarak işleyebilir.

Yüzey geri çatma ile elde edilen yüzey, üçgen veya poligon mesh olarak saklanabilir. Gösterim kolaylığı ve işleme algoritmalarının yaygın olmasından dolayı üçgensel örme yüzeyler yaygın olarak kullanılır. Şekil 2.2'de üçgen örme yüzey saklama amaçlı olarak kullanılan STL mesh formatı gösteriliyor.



Şekil 2.2. STL formatı, StereoLithography Interface Specification, 3D Systems, Inc

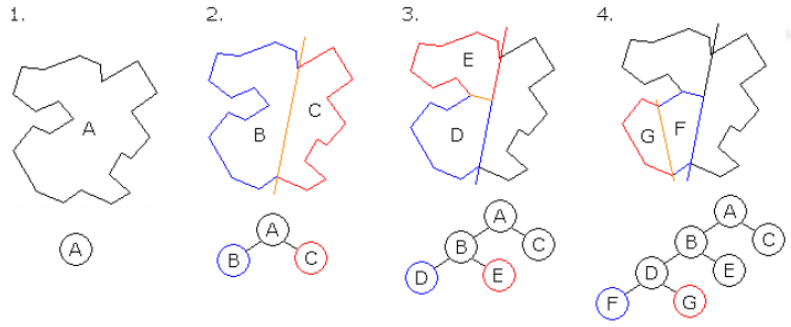
Yüzey geri çatma algoritmasının metotlarının ve tarayıcı cihaza bağlı dış bağımlılıklarının yanı sıra, kullanılan algoritmanın hızı da önemlidir. Delaunay [6] ve Voronoi [7] temelli global yaklaşımlı algoritmalar, genel olarak $O(n^2)$ zaman gereksinimine sahiptir. Yerel yaklaşımlı algoritmalarda, zaman gereksinimi doğrusal düzeydedir. İlerleyen kısımlarda mevcut örme yüzey üretme algoritmalarının temel prensipleri ve karşılaştırılmaları veriliyor.

2.2. Uzayda Hiyerarşik Bölümleme

Grafik uygulamalarında, uzay hiyerarşik bölgelere ayrılarak, grafik elemanları bu bölgeler ile ilişkilendirilir. Bilgisayar grafiği uygulamalarında, modelleme, boyama, animasyon, sanal gerçeklik ve diğer tüm aşamalarda, bu özel bölümleme veri yapılarına ihtiyaç duyulur. Nokta bulutundan yüzey geri çatma uygulamalarında, her bir nokta, hiyerarşik yapıda, bir hacimsel bölge ile ilişkilendirilir. Bu ilişkilendirme, noktalar arasındaki komşulukları bulmak için, bir bölgedeki noktaları belirli bir fonksiyon ile toplu olarak ifade etmek için, veya çıkışta o bölgeden poligon elde etmek üzere kullanılır. Başlıca hiyerarşik uzay bölümleme veri yapıları şu şekilde özetlenebilir:

2.2.1. Bsp-tree

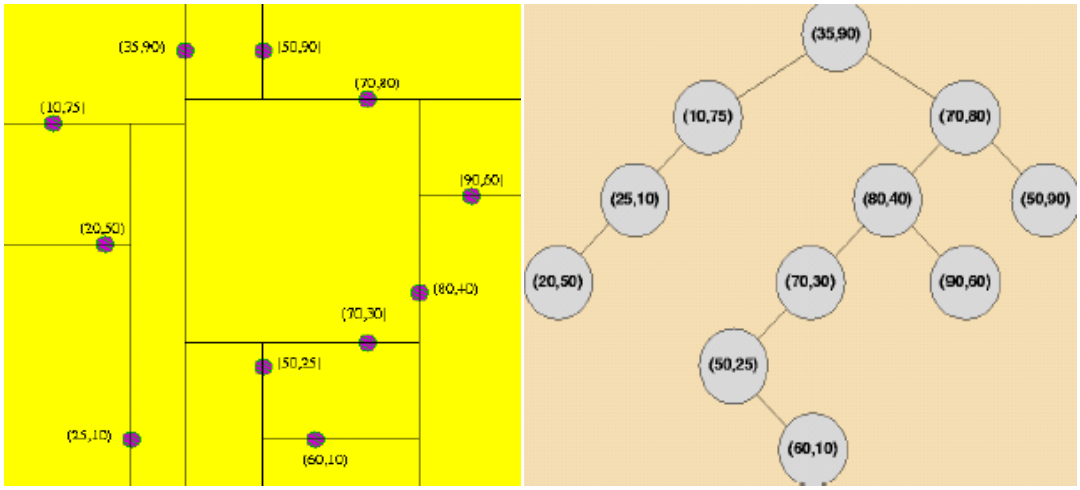
İkili uzay bölme – binary space partitioning algoritmasında, her bir seviyede, uzay seçilen bir düzlemle ikiye bölünür. Şekil 2.3'de BSP-Tree hiyerarşik ağaç yapısı gösteriliyor.



Şekil 2.3. BSP-Tree hiyerarşik ağaç yapısı

2.2.2. kD-tree

K boyutlu ağaç yapısı, kD-Tree yapısı, üç boyutlu uzayda, en yakın komşunun bulunması, k adet en yakın komşuların bulunması ve benzeri amaçlarla kullanılır. Şekil 2.4'de kD-Tree ağaç yapısı ile oluşturulan bir arama veri yapısı gösteriliyor. Şekil 2.5'deki kod parçasında, temel seviyede, hazır verilerin bir kD-Tree yapısına yerleştirilmesi için gerekli kod parçası verilmiştir.



Şekil 2.4. kD-Tree oluşturma

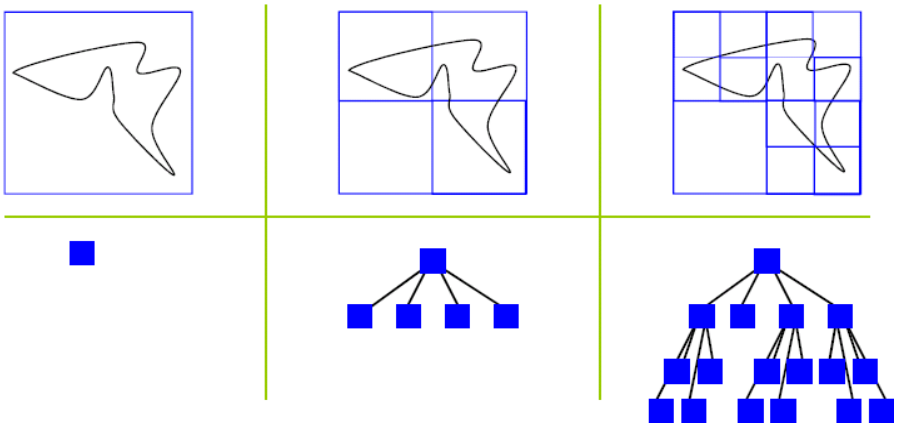
```
        KDNode buildTree(SortedArray S, int cd) {
               if (S.empty()) return(null)
               else if S.singleton() return new
        KDNode(S[x][0],cd ) ;
               else {
                      m=median(S,cd) //median(cutting
               dimension)
                      left = leftPoints(S,cd);
                      right=S-left;
                      t=new KDNode(m) ;
                      t.left=buildTree(left,cd+1);
                      t.right=buildTree(right,cd+1);
               return t
               }
}
```

Şekil 2.5. kD-Tree yapısına yerleştirilmesi için gerekli kod yapısı

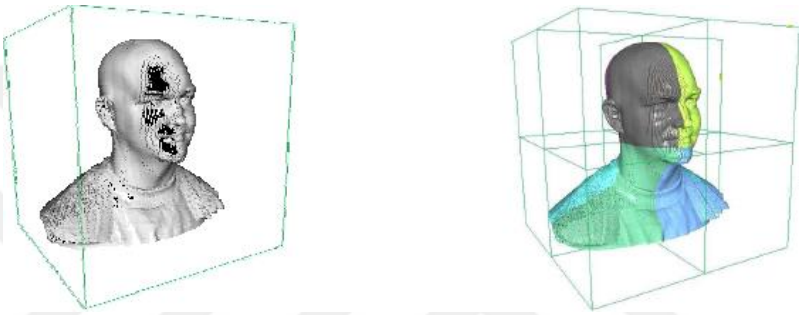
2.2.3. Quadtree/octree

Dörtlü bölmeleme yapısı, iki boyutlu uzayda, alanı hiyerarşik bölgelere ayırır ve bu bölge içinde kalan veri, bölge ile ilişkilendirilir. Bir veri, herhangi bir derinlikteki düğüm ile ilişkili olabilir. Aynı derinlikteki düğümlerden sadece biri ile ilişkili olabilir. Şekil 2.6'da dörtlü ağaç yapısı ile hiyerarşik bölümleme veri yapısı gösteriliyor.

Üç boyutlu uzayda, sekizli ağaç yapısı, Octree kullanılır. Her bir düğüm, sekiz alt parçaya bölünerek ağaç yapısı kurgulanır ve veri, her derinlik seviyesinde bir düğüm ile ilişkilendirilir. Şekil 2.7'de sekizli ağaç yapısı ile uzay bölümleme gösteriliyor.



Şekil 2.6. QuadTree: dörtlü ağaç ile uzay bölümleme



Şekil 2.7. Octree: Sekizli ağaç yapısı ile bölümleme

Şekil 2.8'deki örnek kod parçasında, öz yinelemeli sekizli ağaç yapısının basit bir kod örneği verilmiştir.

```
OctreeNode {
        OctreeNode children[8];
Data data; } // Üst seviye linki + Veri: noktalar, renk vs.
OctreeNode buildOctree (Data data) {
        OctreeNode node;
        if (stopCriteria (data))
                init (node, data);
        else
                Data * childData[8];
                dataSpatialSplit (data, dataChild); //dış kübü
        8'e böl, dağıt
                for (int i = 0; i < 8; i++)
                        node.children[i] = buildOctree
                (childData[i]);
                        node.data = NULL;
return node; }
```

Şekil 2.8. Öz yinelemeli sekizli ağaç yapısının basit bir kod yapısı

2.2.4. Çevreleyen hacim yapısı, bounding 13artez

Çevreleyen hacim yapısında, bir seviyedeki hacim elemanlarının bölgeleri birbirleri ile kesişebilir. Her bir derinlikte, hacim elemanı, birden çok alt parçalara ayrılarak, bu işlem istenen seviye kadar tekrarlanır. Bounding spheres – çevreleyen küreler, bounding box – çevreleyen kutular şeklinde yapılandırılabilir. Şekil 2.9'da çevreleyen küreler yapısı ile işleyen bir yüzey geri çatma uygulaması gösteriliyor.

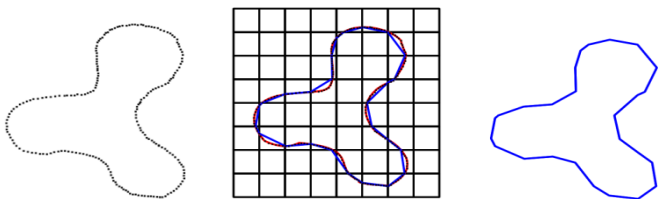


Şekil 2.9. Çevreleyen küreler yapısı ile işleyen bir yüzey geri çatma uygulaması

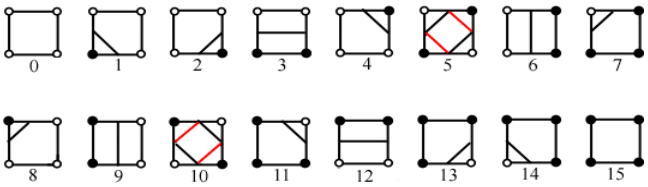
2.3. Sınırlayan Küpler Algoritması – Marching Cubes

Sınırlayan küpler – marching cubes algoritması, voksel (birim hacim elemanı) uzayındaki verilerin poligon olarak geri çatılması için, veya parametrik olarak ifade edilen yüzeylerin poligonlaştırılması için kullanılabilir.

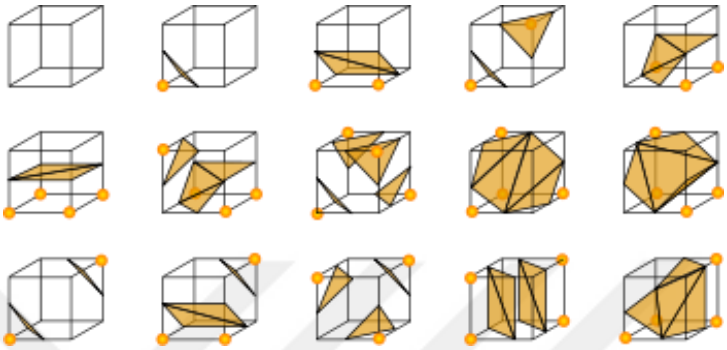
Sınırlayan küpler – marching cubes algoritması, 1987 yılında Lorensen ve Cline [4] tarafından, tomografi verilerinin görüntülenmesi amacı ile geliştirilmiştir. Giriş uzayı küplere bölünerek, her bir hacim elemanı için, önceden tanımlanmış belirli küp kombinasyonlarından, yüzeyi en iyi sınırlayanı seçilerek, yüzeyi temsil eden poligon örme yüzey elde edilir. Şekil 2.10 ve 2.11'de algoritmanın iki boyutlu bir gösterimi verilmiştir. Şekil 2.12'de üç boyutlu durumlar gösteriliyor. Şekil 2.13'de ise bir MR verisi üzerinde sınırlayan küpler algoritması ile elde edilen mesh gösteriliyor.



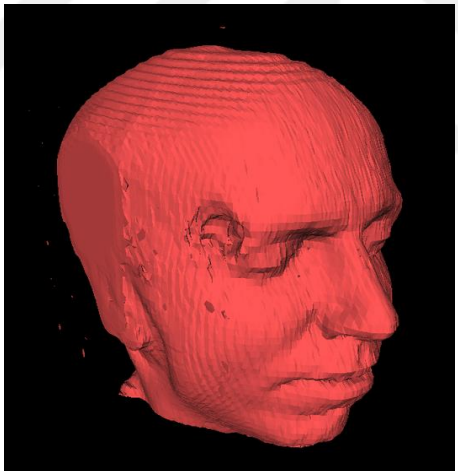
Şekil 2.10. Sınırlayan kareler



Şekil 2.11. Sınırlayan kareler için kombinasyonlar [4]



Şekil 2.12. Sınırlayan küpler için kombinasyonlar Simetriler ile birlikte 256 durum oluşur [4]



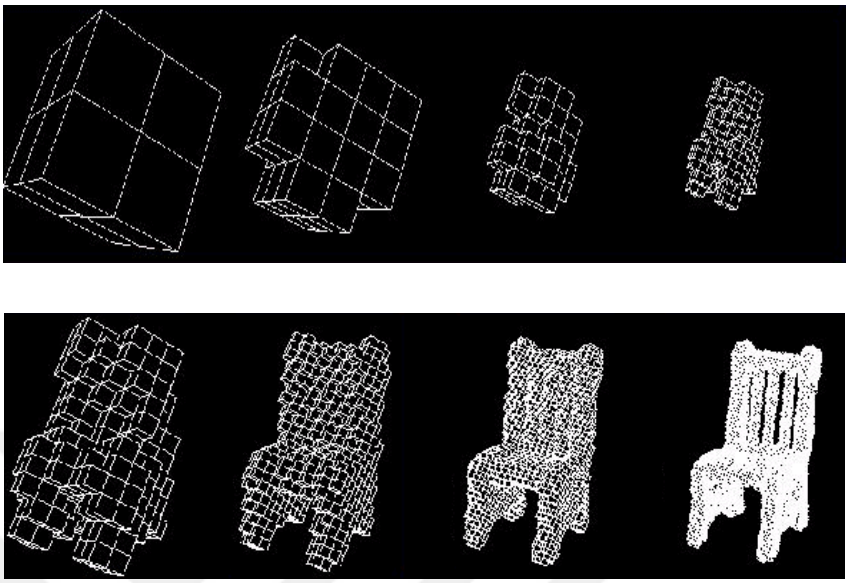
Şekil 2.13. Sınırlayan küpler, 150 dilimli MR verisi üzerinden çıkarım

2.4. Space Carving – Boşluk Yontma

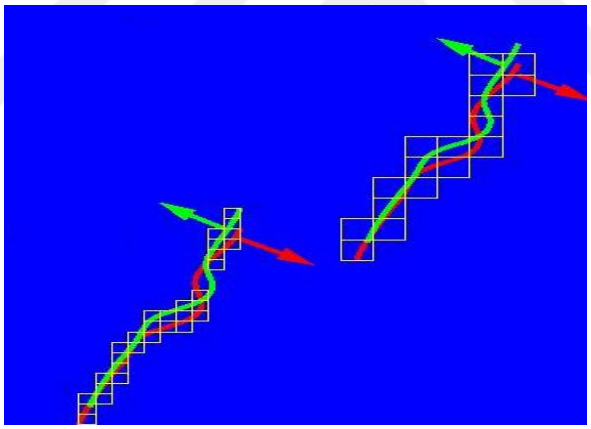
Kari Pulli'nin çalışmasında [17], veriyi tümüyle içine alan bir küpten yola çıkılarak, bu küp hiyerarşik olarak sekizli alt parçalara ayrılmakta ve veri içermeyen kısımlarçöpe atılarak, veriyi içeren sınırlar elde edilmektedir. Şekil 2.14 ve 2.15'de bu algoritma görsel olarak verilmiştir.

– Hacim iki düzlem ile sekiz parça yapılır – Octree yapısı

– Bu işlem her bir parça küp için tekrar edilir
– Dış bölgede kalan küpler elenir



Şekil 2.14. Boşluk yontma algoritmasının aşamalı gösterimi [17]



Şekil 2.15. İki boyutlu gösterim [17]

2.5. Örme Yüzey Üretme Algoritmaları

Laser tarama cihazlarının gelişmesi ile birlikte, sahadan toplanmış nokta bulutu verisinden yüzey geri çatma problemi, yoğun bir araştırma konusu olmuştur. Problem ilk olarak 1980'lerin ortalarında, Boissonnat [8] tarafından tanımlanmıştır. 1990 yılında, Hoppe ve ark. [3] tarafından yapılan çalışma, düzensiz nokta bulutu verilerinden yüzey geri çatmada ilerleyen bir aşama olmuştur.

Yüzey geri çatmada araştırmalar temel olarak iki yönde ilerlemiştir. Hesaplamalı geometri metotları olarak anılan, Delaunay [6] üçgenleme tekniği kullanılan

çalışmalarda [10 – 16] hesaplamalar Delaunay kompleksine dayandırılır ve nokta verisi üzerinde Voronoi [7] diyagramı kullanılarak ikili – dual graph yapılar bina edilir [18, 19].

Yüzey geri çatmada diğer bir trend, hacimsel – volumetrik metotlar [20 – 23] olmuştur. Bu metotlarda, nokta bulutu verisi belirli parçalara ayrılarak bunlara üç boyutlu fonksiyonlar ile yaklaşımlar üretilir. Bu fonksiyonlardan, sıfır seviyesi (zero-level set) çıkarımı ile, sınırlayan küpler – marching cubes veya benzeri algoritmalar kullanılarak, yüzey açık olarak elde edilmeye çalışılır. Bu konuda, MLS (moving least squares) yüzeyler [24 – 27] üzerinde çalışmalar olmuştur. Ayrıca Radyal Tabanlı Fonksiyonlar da (RBF) benzer şekilde kullanılmıştır [21].

Tarama cihazlarından elde edilen verilerin gürültülü olması sebebiyle, yüzey geri çatma algoritmalarının, gürültüye dayanıklı olması, istenen bir özelliktir. [28, 29, 30, 31] no'lu çalışmalarda, gürültülü ve eksik veri üzerinde işleyen çeşitli algoritmalar geliştirilmiştir. Tarama cihazının yüzeyin belirli parçalarında az yoğun örnek alması neticesi, eksik ve homojen olmayan veriler üzerinde de çalışabilen algoritmalara ihtiyaç duyulmuştur [32 – 35]. Mevcut yüzey geri çatma algoritmalarının geniş bir özeti [36] no'lu çalışmada bulunabilir.

Aşağıdaki listede, yüzey geri çatmada kullanılan yerel ve global algoritmaların bir sınıflandırılması yapılmıştır [36, 39].

–Global Yaklaşımlar, Delaunay ve Voronoi Bazlı Algoritmalar (Hesaplamalı Geometri Metotları)

- oCrust [10]
- oCocone [15]

–Yerel – Lokal Yaklaşımlar (Volumetrik metotlar)

- oK-En yakın komşular
- oTemel Bileşenler Analizi – Principle Component Analysis – PCA
- oGrid Bazlı Kesikli (Discrete) Algoritmalar
  - ▪Tanjant Düzlem Yaklaşımı [3]
  - ▪Top Yuvarlama – Ball Pivoting [13]
- oImplicit (Kapalı) Surface Algorithms
  - ▪MLS, Moving Least Squares [24]

▪Partition of Unity (PU) [22]

▪Radyal Tabanlı Fonksiyonlar [21]

Bu algoritmaların özetleri ilerleyen başlıklar altında anlatılıyor.

2.6. Delaunay Temelli Algoritmalar

Hesaplamalı geometri alanında, Delaunay üçgenlemesi $DT(P)$, bir P noktalar kümesi için düzlemde şu şekilde tanımlanır: P kümesindeki hiçbir nokta, diğer üçgenlerin teşkil ettiği çevreleyen çemberlerin içinde olmamalıdır. Şekil 2.16'da örnek bir delaunay üçgenlemesi gösterilmektedir. Köşeler P kümesini teşkil ediyor. Bu metot 1934 yılında, Boris Delaunay [6] tarafından geliştirilmiştir.



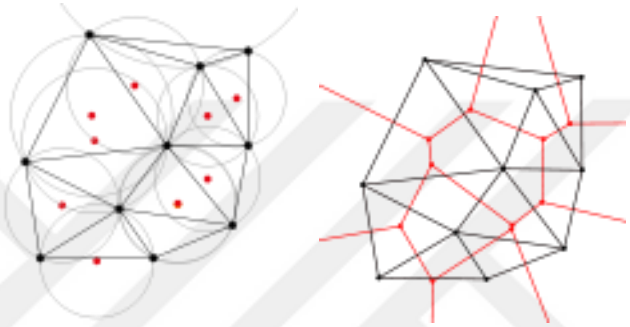
Şekil 2.16. Delaunay üçgenleme

P nokta kümesinden üretilen $DT(P)$ Delaunay üçgenlerinin oluşturduğu yapı, P nokta kümesinin Voronoi [7] diagramının eşlenik yapısına (dual graph: her yüz için bir köşe sahibi) denk gelir. Şekil 2.17'de bu durum gösterilmektedir.

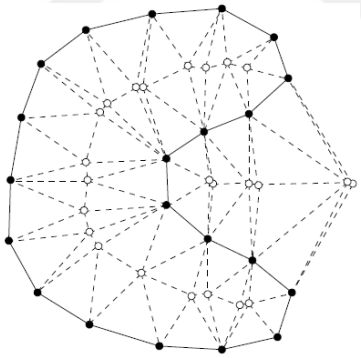
Noktalar arasındaki komşuluk ilişkilerinin Delauay üçgenlemesi ile bulunmasından sonra, bu kümenin çeşitli alt kümeleri kullanılarak, yüzey geri elde edilmektedir. Crust, Cocone [10, 11, 15, 16] ve benzeri algoritmalar, hesaplamalı geometri tekniklerini kullanarak benzer şekilde çalışmaktadır.

Amenta ve Bern [10] tarafından geliştirilen yüzey geri çatma amaçlı Crust algoritması, Delaunay ve Voronoi metotları ile işleyen algoritmalardan biridir. Yüzeyi tanımlayan S nokta kümesi için $DT(S)$ hesaplanır. Voronoi diyagramı

$V(S)$'in kutup noktaları Q nokta kümesini teşkil eder. Son olarak, S ve Q noktalarının birleşim kümesi kullanılarak, bu kümenin delaunay üçgenlemesi $DT(S \bigcup Q)$ üretilir [36]. Şekil 2.18'de crust algoritmasının parçaları gösterilmektedir. Siyah noktalar S örneklenmiş nokta kümesini, beyaz noktalar ise Q Voronoi kutuplarını göstermektedir. Crust kümesi, sürekli çizgiler ile gösterilmiştir [9].



Şekil 2.17. Delaunay üçgenleme ve karşılık voronoi diyagramı [6]

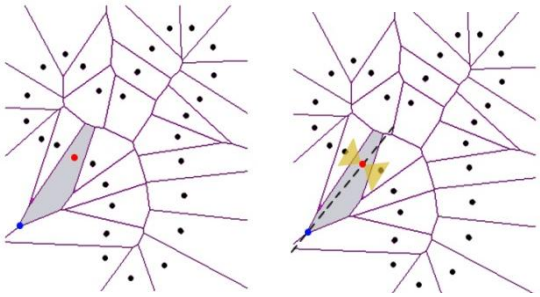


Şekil 2.18. Crust algoritması

Cocone [15, 16] algoritması ise, Crust algoritmasının devamı olarak, Amenta, Choi, Dey ve Leekha tarafından geliştirilmiştir. Şekil 2.19'da cocone algoritmasından bir resim gösterilmektedir.

2.7. Yerel Bazlı Algoritmalar

Nokta bulutundan yüzey geri çatma probleminde, ilk yaklaşımlar, noktaların komşuluk ilişkilerinin kullanılması şeklinde olmuştur.



Şekil 2.19. Cocone algoritması [16]

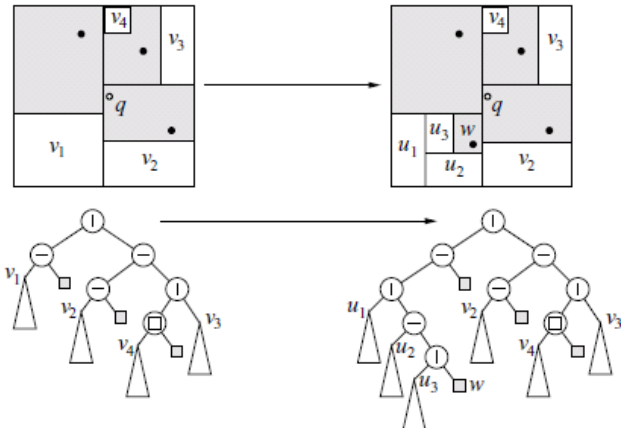
Her bir nokta için, yüzey normalinin belirlenmesinde, k adet en yakın komşular veya belirli bir mesafedeki tüm komşular kullanılır [5]. Seçilen noktaların belirlediği düzlem ve bu düzleme ait normalin yönü, poligon geri çatmada referans olarak kullanılır.

2.7.1. k-en yakın komşuların bulunması

Üç boyutlu nokta verilerinde, her bir nokta için, o noktaya en yakın k adet komşu noktanın bulunması, $O(n^2)$ zamanda, kaba hesapla bulunabilir, ancak bu çözüm çok fazla sayıda noktanın bulunduğu veri kümelerinde uygulanabilir bir çözüm üretmez. Noktaların, hiyerarşik bir ağaç yapısına dağıtılması ile, en yakın komşuların bulunması işlemi her bir arama için $O(n \log n)$ zamanda tamamlanabilir. Üç boyutlu uzayda nokta arama amaçlı olarak Arya ve ark. [37] tarafından geliştirilen balanced box decomposition *(BDD)* ağaç yapısı, hiyerarşik uzay bölümlemesinde, ağaç yapısının derinliği dengelenerek, üç boyutlu aramalar, $O(3\log n)$ zamanda yapılabilir [39]. Ağaç yapısı ile sağlanan arama fonksiyonu kullanılarak, nokta bulutundaki her bir noktanın, m adet komşusu bulunur ve bu komşuları temsil eden bir düzlem ve normal hesaplanır.

2.7.2. Temel bileşenler analizi (principle component analysis, pca)

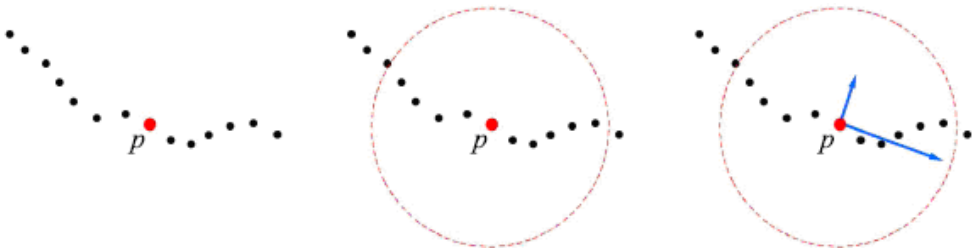
Temel bileşenler analizi ile, uzaydaki nokta kümesi, yeni bir uzayda, n adet eksen üzerinde yerleştirilir. Yeni uzayı belirleyen, birbirine dik vektörler, eigen vektörler olarak anılır. Her bir eigen vektörün ilişkilendirildiği bir eigen değeri vardır. Bu değer, eigen vektörün gösterdiği yönde, nokta bileşenlerin saçılmasını ifade eder.



Şekil 2.20. BDD tree yapısı

Bir noktanın, m adet komşularının, (x, y, z) 20artezyen uzaydaki koordinatları girdi olarak kullanılarak, temel bileşenler analizi ile, o noktanın komşuları arasındaki saçılmanın yönü bulunur. Sahadan toplanmış olan nokta verilerinin düzenli dağıldığı farz edilerek ve bir noktaya ait komşuların aynı düzlemde olduğu farz edilerek, noktaya ait normalin yönünün, en küçük eigen değerine sahip eigen vektör olduğu sonucuna varılır. Çünkü noktaya ait normalin yönünde hiçbir saçılma beklenmez, saçılmaların, normalin dikinde bulunan tanjant düzlem üzerinde olması beklenir. Böylece, normalin dışında kalan iki eigen vektör, noktanın üzerinde bulunduğu düzlemi tanımlamış olur. Şekil 2.21'de iki boyutlu olarak, p noktasına ait komşular ve bu komşulara ait eigen vektörler gösteriliyor.

Eigen vektörlerin yönü, saçılmanın yönünü gösterirken, vektörlere ait eigen değerleri, saçılmanın genliğini gösterir. En az saçılmanın olduğu eigen vektör, noktaya ait normalin yönü olarak farz edilir [39].



Şekil 2.21. P noktası ve yakın komşuları için temel bileşenler analizi

Şekil 2.21 için, N_p ifadesi, p noktasına ait k adet en yakın komşular kümesini göstersin. O_p ifadesi, N_p kümesindeki noktaların orta noktasını göstersin. 3*x*3 boyutlu simetrik kovaryans matrisi şu şekilde tanımlanır:

$$Kovar\ yansMatris = \sum_{x \in N_p} (x - o_p) \otimes (x - o_p) \tag{2.1}$$

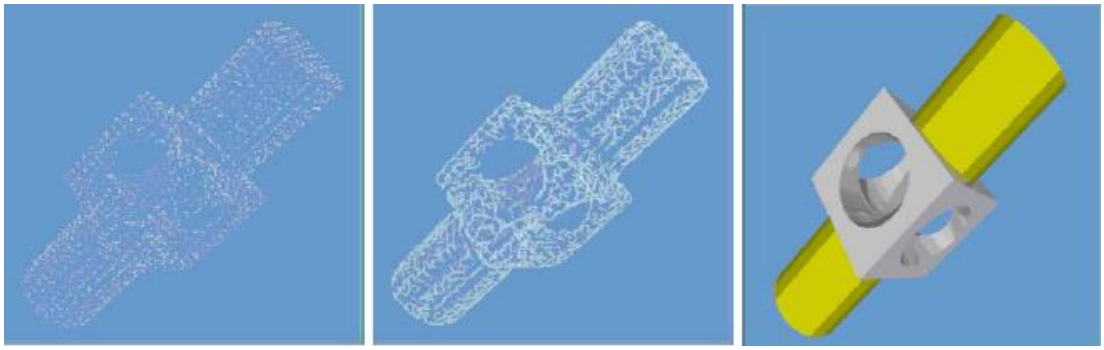
Buradan, eigen vektörler ve eigen değerleri hesaplanır [39]. Burada $\otimes$ ifadesi dış çarpımı gösterir: $u \otimes v = uv^T$.

2.7.3. Yerel – grid bazlı algoritmalar

Bu bölümdeki algoritmalar, noktaların komşuluk ilişkileri kullanılarak işler. 1990'li yılların başlarında, nokta bulutu verisi işlemede kullanılmaya başlanan ilk yöntemlerdir.

2.7.3.1. Tanjant düzlem yaklaşımı

Nokta bulutundan yüzey geri çatmada, ilk çözümlerden biri, 1990 yılında Hoppe ve ark. [3] tarafından yapılan çalışma olmuştur. Benzer metotları içeren diğer bir çalışma 1996 yılında Curles ve Levoy [40] tarafından yapılmıştır. Hoppe'nin çalışmasında, öncelikle her bir nokta için temel bileşenler analizi kullanılarak tanjant düzlem yaklaşımları bulunur. Komşu noktaların ilk aşamada hesaplanan tanjant düzlemlerin uyumluluğunun sağlanması için, ilave bir düzeltme metodu uygulanır. Şekil 2.22'de Hoppe'nin çalışmasındaki resim gösteriliyor.



Şekil 2.22. Hoppe mesh üretme algoritması [3]

Algoritma şu şekilde özetlenebilir:

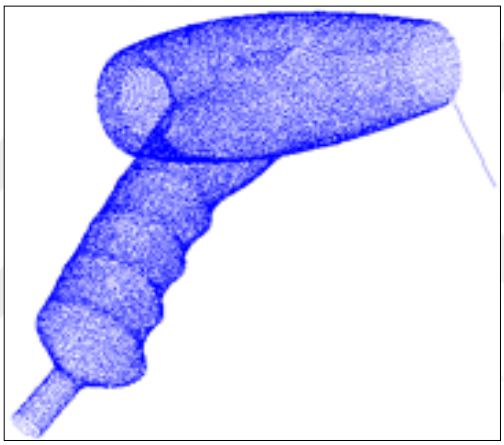
- 3D noktayı, bilinmeyen yüzeye eşleştirilen işaretli mesafe fonksiyonunu belirlenir. Her bir nokta için, işaretli mesafe fonksiyonu, yaklaşık olarak bulunan tanjant düzleme olan uzaklığını ifade eder. Mesafenin işaretli olması, noktanın, düzlemin hangi tarafında olduğunu gösterir

(a) Her bir nokta için tanjant düzlem yaklaşımını bulunur

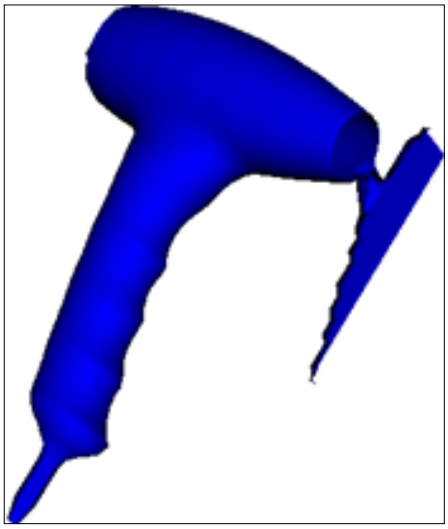
(b) Tanjant düzlemleri, komşuları ile kıyas ederek düzeltilir

- Sınırlayan küpler (marching cubes) algoritması kullanarak yüzey çıkartılır

Şekil 2.23 ve 2.24'de VTK kütüphanesi kullanılarak, Hoppe'nin metodu ile nokta bulutundan çıkarılmış örme yüzey gösteriliyor.



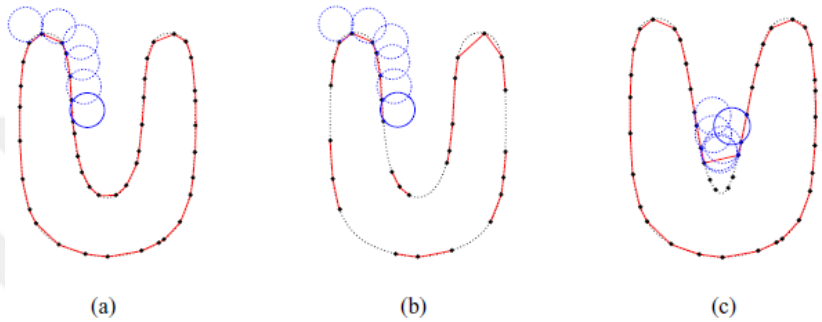
Şekil 2.23. VTK kütüphanesi ile nokta bulutu gösterimi



Şekil 2.24. Nokta bulutundan elde edilen örme yüzey

2.7.3.2. Top yuvarlama – ball pivoting algoritması

Top yuvarlama – ball pivoting algoritması Bernardini ve ark. [13] tarafından geliştirilmiştir. Örnek noktaların arasından tümüyle geçemeyecek büyüklükteki ρ yarıçaplı bir top, üç noktaya dokunarak döndürülmeye başlanır. İki noktası sabit kalmak üzere temas edilen üçüncü noktala ile eski iki temas noktası bir mesh üzerinde bir üçgen teşkil eder. Top bu şekilde tüm veri noktaları üzerinde dolaştırılır. Şekil 2.25'de top yuvarlama algoritmasının iki boyutlu benzetimi gösteriliyor.



Şekil 2.25. Ball pivoting algoritması

2.7.4. Kapalı Fonksiyon Yüzey Algoritmaları

Kapalı (implicit) fonksiyon kullanılan algoritmalarda, yaklaşım yapılacak olan yüzeyin verileri kapalı (implicit) fonksiyonlar ile ifade edilmeye çalışılır [9].

Kapalı fonksiyonlar cinsinden yüzey geri çatma problemi şu şekilde tanımlanabilir: Kübik uzayda tanımlı S yüzeyi üzerinde n adet farklı nokta verilmiş olsun. S yüzeyine yaklaşım yapan bir S' yüzeyi bulunması istenir. Yaklaşım yüzeyini bulmak için, kapalı fonksiyon kullanan metotlarda, fonksiyonun sıfır olduğu (zero level set) noktalarda, S yüzeyine yaklaşım yapan ve enterpole eden bir $y = f(x)$ fonksiyonu aranır [39]. Geri elde edilen S' yüzeyi, S yüzeyine bir yaklaşım olduğundan, veri noktaları tam olarak geri elde edilen yüzey üzerinde bulunmaz.

MLS (moving least squares) ve RBF (radial basis functions) metotları, bu metotlar arasında sayılabilir.

2.7.4.1. Moving least squares

Moving least squares (MLS) yüzeyleri, komşu nokta noktaların oluşturduğu hacimsel bir kümenin bir düzlem üzerine projeksiyonu ile elde edilir. Komşuluk genellikle girdi veri noktası p_i merkezli topların birleşimi olarak tanımlanır. Geleneksel projeksiyon metodu aşırı hesaplama yükü getirdiğinden dolayı, Adamson ve Alexa [24] tarafından tanımlanan daha basit bir projeksiyon metodu kullanılır. Tekrarlı olarak, x noktasının ve $n(x)$ normalinin, komşu noktaların ağırlıklı ortalamaları tarafından tanımlanan bir yüzeye izdüşümü alınır [9].

$$a(x) = \frac{\sum_j \theta\left(\left\|x - p_j\right\|\right) p_j}{\sum_j \theta\left(\left\|x - p_j\right\|\right)} \tag{2.2}$$

$$n(x) = \frac{\sum_j \theta\left(\left\|x - p_j\right\|\right) n_j}{\left\|\sum_j \theta\left(\left\|x - p_j\right\|\right) n_j\right\|} \tag{2.3}$$

Girdi veri kümesinin normallerinin bilinmemesi durumunda, $n(x)$ normal değerleri, PCA metodu ile komşu veri noktaları kullanılarak bulunur.

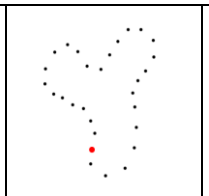
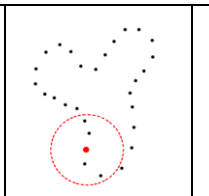
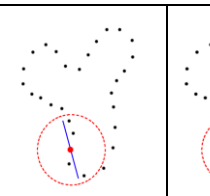
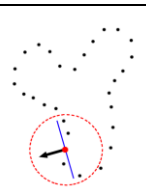
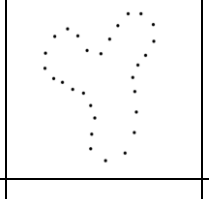
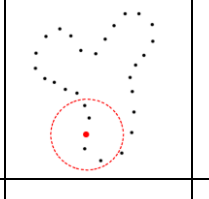
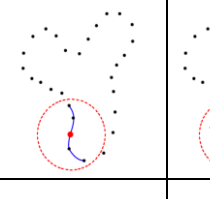
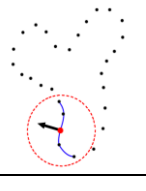
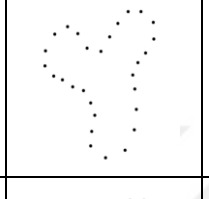
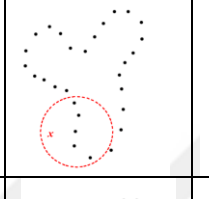
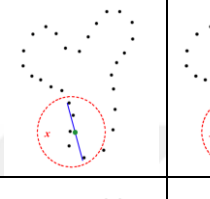
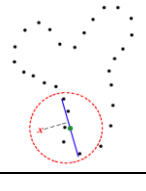
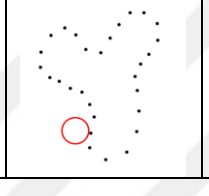
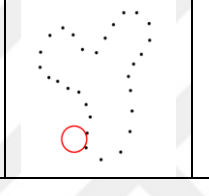
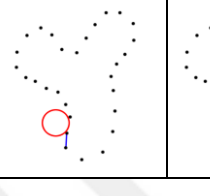
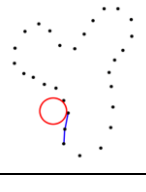
2.7.4.2. Radyal tabanlı fonksiyonlar

Radyal tabanlı fonksiyonlar (RBF), kapalı fonksiyon kullanan yüzey geri çatma algoritmalarında sıklıkla kullanılmaktadır. Verilen bir nokta kümesi için, sıfır seviyeleri (zero level set) bu kümeye yaklaşım yapan veya enterpole eden bir bileşik fonksiyon $f: R^3 \rightarrow R$, daha basit fonksiyonların kombinasyonu olarak tanımlanır: [9, 39]

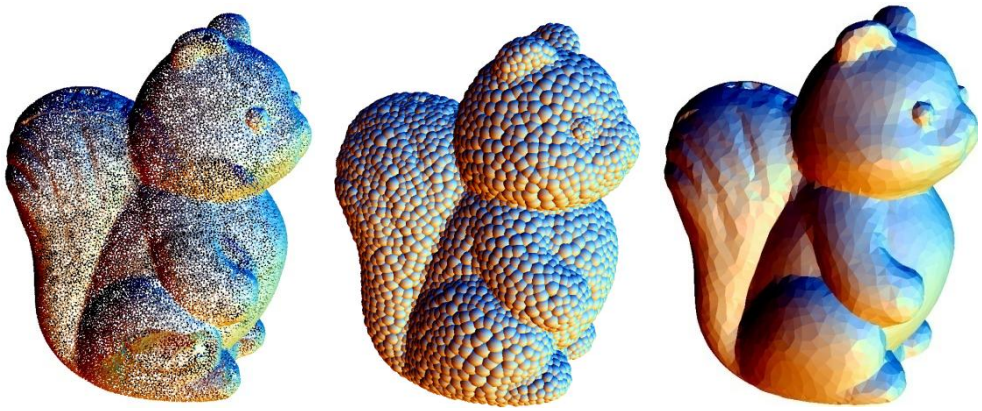
$$f(p) = \sum_{i=1}^{m} \alpha_i \phi\left(\left\|x - c_i\right\|\right) \tag{2.4}$$

Bu denklemde α_i bilinmeyen ağırlıkları, c_i radyal fonksiyonun merkezini göstermektedir. $\phi(r)$ radyal fonksiyonu temsil eder ve genellikle, $\phi(r) = r^3$, $\phi(r) = r^2 \log(r)$ benzeri radyal fonksiyonlar kullanılır [9]. Tablo 2.1'de çeşitli yüzey geri çatma metotlarının iki boyutlu modeli gösteriliyor [39].

Tablo 2.1. Çeşitli yüzey geri çatma metotların iki boyutlu gösterimi

Tanjant Düzlem Yaklaşımı				
Radyal Tabanlı Fonksiyonlar				
Hareketli En Küçük Kareler (MLS)				
Ball Pivoting				

Şekil 2.26'da Ohtake'nin [22, 31] çalışmasındaki RBF metodu ile nokta bulutundan örme yüzey geri elde etmedeki aşamalar gösteriliyor.



Şekil 2.26. Ohtake, nokta bulutu işleme aşamaları

2.8. Metotların Karşılaştırılması

Tarama ortamından gelen gürültülü ve yoğun veriler ile çalışabilmesi ve üssel olmayan zaman gereksinimi ile işleyebilmeleri, radyal fonksiyon temelli algoritmaları son yıllarda ön plana çıkarmıştır. Bununla birlikte, radyal tabanlı fonksiyonlar ile üretilen örme yüzey yapılarında, köşelerin orijinal yüzey üzerinde

bulunmaları garanti edilmez. RBF metodunda, veriler üzerinde yumuşatma ve filtreleme gerçekleştirilir. Tablo 2.2'de bazı metotların gürültülü veri davranışı, zaman gereksinimi ve köşelerin yüzeyin üzerinde olması açısından karşılaştırması veriliyor [39].

Tablo 2.2. Metotların karşılaştırılması

Algoritma	Gürültülü Veri	Zaman	Yüzeyin Üzerinde Köşeler
Tangent Plane			Evet
Triangle Fans			Evet
MLS	Evet		Evet
RBF	Evet		
Iso Surface	Evet		
Crust		Üssel	
Cocone		Üssel	Evet

3. ÖRME YÜZEY SADELEŞTİRME ALGORİTMALARI

Örme yüzeyler bilgisayarlı grafik uygulamalarında çevreleyen yüzeyi temsil eden temel veri yapılarındandır. Örme yüzey poligonları, üçgenler, dörtgenler veya diğer çokgenlerden oluşabilir. Özel donanımlı grafik kartları kullanılarak hızlı bir şekilde ekran görüntüsü halinde gösterilebilirler.

Tarama cihazlarından gelen yoğun ve düzensiz veri, nokta bulutundan mesh aşamasına geçerken, çok sayıda gereksiz poligon üretilmesine neden olabilir. Benzer şekilde bilgisayarlı tomografi ve diğer tarama cihazlarından elde edilen poligon yüzey, çevreleyen yüzeyi temsil etmenin haricinde çok sayıda gereksiz poligon içerir.

Nokta bulutu verilerinin temsili fonksiyonlar kullanılarak ifade edilmesi ve sonrasında yüzey çıkartma metotları ile (isosurface extraction: marching cubes v.b.) örme yüzey elde edilmesi aşamasında, çok sayıda ve tekrarlı, istenmeyen kenar-köşe ve yüz parçaları oluşabilir. Mesh sadeleştirme metotları kullanılarak, örme yüzeyin temizlenmesi ve yüzey topolojisi bozulmadan daha az poligon ile yüzeyin ifade edilmesi istenir.

Sadeleştirme neticesinde elde edilen poligon verisi, aynı modeli, topolojisi bozulmadan daha az sayıda veri kullanarak ifade etmeyi amaçlar. Bu şekilde, örme yüzeyin bilgisayar ortamında hızlı ve verimli bir şekilde gösterilmesine imkan sağlanır. Grafik donanım gereksinimi ve elektrikli - manyetik hafıza gereksinimi düşer. Grafik modellerin ağ ortamında kullanılması durumunda, ağ trafiği gereksinimi azalır.

Mesh sadeleştirme yöntemlerinden, iletişim, tıbbi görüntüleme, bilgisayar destekli tasarım, reklamcılık, animasyon ve bilimsel veri gösterimi alanlarında yararlanılmaktadır [41].

Poligon basitleştirme işlemi, model basitleştirme işlemlerinin yanı sıra, değişik detay seviyeleri (level of details: LOD), kademeli transfer ve model sıkıştırma işlemlerinde

de kullanılır. Model basitleştirme, bir poligon modelin, dış görünümünü değiştirmeden, daha az poligon ile ifade edilmesini amaçlar. Bu şekilde grafik gösterimde performans artışı hedeflenir. Model basitleştirme işleminde, kullanıcı ile interaktif bir şekilde sadeleştirme yapmanın ötesinde, sadeleştirme işleminin otomatik olarak yapılması hedeflenir.

Oyun ve diğer animasyon - grafik uygulamalarında, bir nesne sahnedeki konumuna göre detaylı veya kaba olarak gösterilir. Bu durumda nesnenin değişik detay seviyelerine (level of details) ait poligon modellerinin hafızada olması gerekir. Detay seviyeleri arasındaki yumuşak gösterimli geçişlerin canlandırılması da ayrı bir problemi teşkil eder.

Poligon modellerin ağ ve interaktif ortamlarda taşınması amacıyla, kademeli transfer tekniği (progressive meshes) kullanılır. Kademeli transfer tekniğinde, modelin basitten komplekse giden aşamaları arasındaki geçişler fonksiyon olarak saklanır. Kullanıcı, ağ üzerinde, modelin basit ve kompleks durumları arasında geçişler yapabilir.

Poligon modellerin saklanması aşamasında, sıkıştırılarak, daha az depolama alanı kullanılması istenir. Model sıkıştırma işlemleri de benzer algoritmalar kullanılarak yapılır.

Mesh sadeleştirme algoritmaları şu şekilde sınıflandırılabilir [42]:

–Yerel Sadeleştirme Algoritmaları

oKöşe Yok Etme (Vertex Decimation)

oKenar Yok Etme (Edge Contraction)

–Global Sadeleştirme Algoritmaları

oKöşelerin Birleştirilmesi (Vertex Clustering)

oModel Yaklaşımı (Shape Approximation)

3.1. Yerel Sadeleştirme Algoritmaları

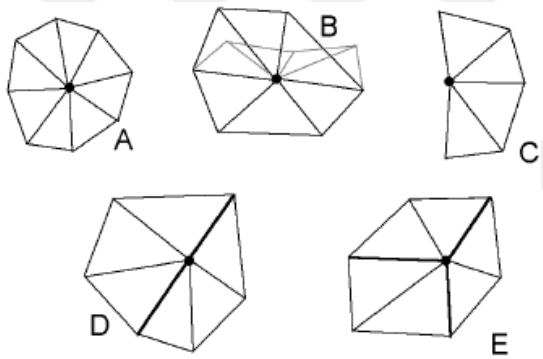
Sadeleştirme yöntemleri genel olarak ikiye ayrılır. Yerel sadeleştirme işleminde, sadeleştirme işlemi sadece bir bölgeye veya seçilen alana uygulanırken, global yöntemlerde, mesh model bir bütün olarak ele alınır.

3.1.1. Köşe yok etme (vertex decimation)

Bu gruptaki algoritmalar, poligonal modelde her safhada iteratif olarak bir poligon köşesini attıktan sonra, bu köşeyi kullanan poligon yüzeyleri de atar ve daha sonra geriye kalan boşluğu tekrar üçgenlere böler [43]. Bu algoritmalar, modellerin topolojisini koruyan algoritmalardır. Aşağıdaki aşamalarda köşe yok etme metodunun detayları anlatılıyor.

3.1.1.1.Köşe sınıflandırma (vertex classification)

Köşe yok etme algoritmasında aşamalar, köşelerin sınıflandırılması ile başlar. Her bir köşe, komşuluk ilişkilerine göre sınıflandırılır [44]. Bu şekilde silinmeye aday köşeler tespit edilmeye çalışılır. Şekil 3.1'de beş çeşit sınıfa ait durumlar gösteriliyor.



Şekil 3.1. Köşe Sınıflandırma [44]

- Basit köşe (A): Tümüyle üçgenler ile çevrili durumdaki bir köşe yapısını ifade eder. Çevreleyen üçgenler özellik kenarı barındırmaz.
- Kompleks Köşe (B): Tümüyle üçgenler ile çevrili durumdaki bir köşedir. Ancak bu köşeyi kullanan kenarlar, birden çok üçgen tarafından paylaşılıyor olabilir.
- Sınırlayan Köşe (C): Yüzeyin sınırında bulunan, yarısı üçgenler ile çevrili köşe.
- Dahili Köşe (D): Bir kenarı paylaşan iki üçgenin normallari arasındaki açı, belirli bir özellik açısı (feature angle)'ndan yüksek ise, bu kenar ayırt edici özellik kenarı (feature edge) kabul edilir. İki adet özellik kenarı ortasında bulunan bir köşe, dahili köşe olarak işaretlenir.
- Tüm Köşe (E): Özellik kenarı üzerindeki bir köşe, birden çok özellik kenarı tarafından kullanılıyor ise, tüm köşe olarak işaretlenir [44].

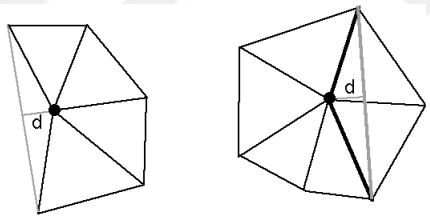
3.1.1.2.Yok etme kriteri (decimation criterion)

Köşelerin sınıflandırılmasından sonra, bu köşenin silinmesi halinde ortaya çıkacak hataya bir yaklaşım yapılır. Hata ölçümü, köşe sınıflandırma bilgisine göre yapılır.

Basit köşeler için hata ölçüsü, tanım gereği çevreleyen üçgenler hiçbir özellik kenarı içermediğinden, çevreleyen üçgenlerin yaklaşık olarak düz bir yüzey üzerinden oldukları farz edilir. Bu çevreleyen üçgen yapısını en az hata ile temsil eden düzlem bulunur. En küçük kareler metodu kullanılır ise, üçgenlerin köşeleri ile düzlem arasındaki mesafenin karesi minimize edilir. Ortalama metodu kullanılır ise, yüzeyin normali, köşeyi çevreleyen üçgenlerin normallerinin ortalaması olarak alınır.

Bulunan düzlem ile yüzey arasındaki mesafe, değerlendirme kriteri olarak kullanılır. Yok etme kriterinin, düz bölgelerdeki köşelerin yok edilmesine öncelik verir iken, özellik kenarları üzerindeki köşelerin yok edilmesini geciktirmesi beklenir.

Sınırlayan köşe ve dahili köşeler için, bu köşe silindiğinde oluşacak yeni kenar ile bu köşe arasındaki mesafe ölçüt olarak kullanılır. Şekil 3.2'de bu durum gösteriliyor.

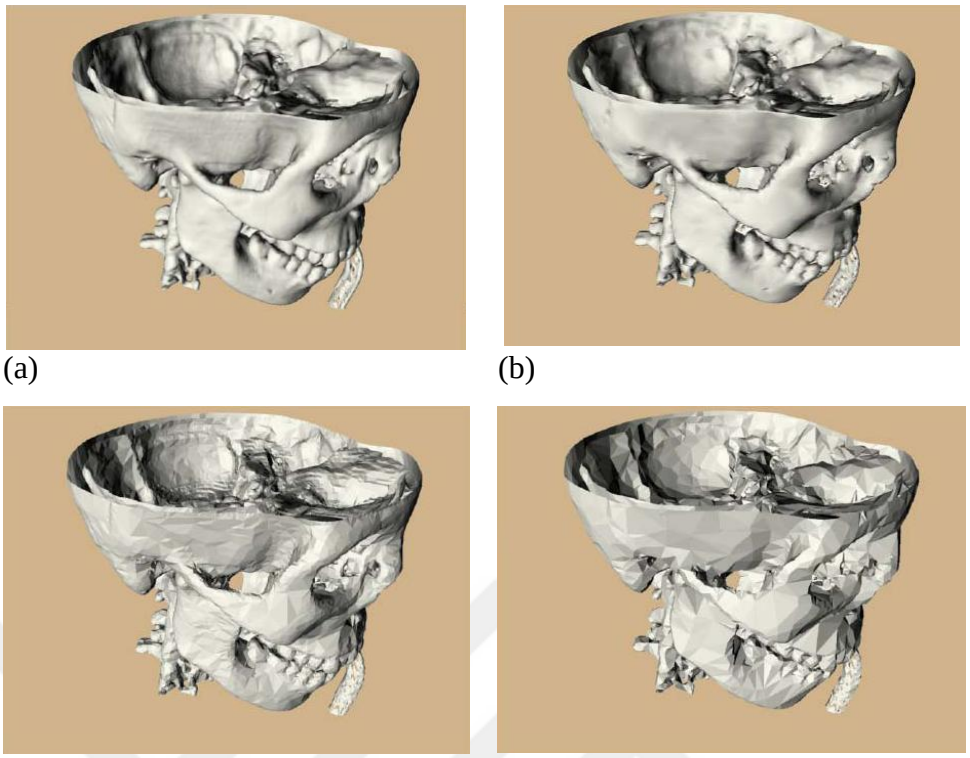


Şekil 3.2. Sınırlayan köşe ve dahili köşe hata ölçütü

Yok etme kriteri olarak kullanılan mesafelerin belirli bir ölçeğin altında kaldığı köşeler, silinmek için seçilmiş köşeler olacaktır. Bu köşelerin silinmesinden sonra ortaya çıkan boşluğun, köşe yok edildikten sonra tekrar üçgenlerle doldurulması gerekir [44].

3.1.1.3.Üçgenleme

Silme için seçilen köşenin yok edilmesinden sonra ortaya çıkan boşluk, tekrar üçgenler ile doldurulur. Şekil 3.3'de VTK kütüphanesi ile yapılan mesh sadeleştirme işlemi gösteriliyor.



Şekil 3.3. VTK kütüphanesi ile yapılan mesh sadeleştirme işlemi [44] a) 569K üçgen b) 142K üçgen c) 142K üçgen, düz d) 57K üçgen, düz

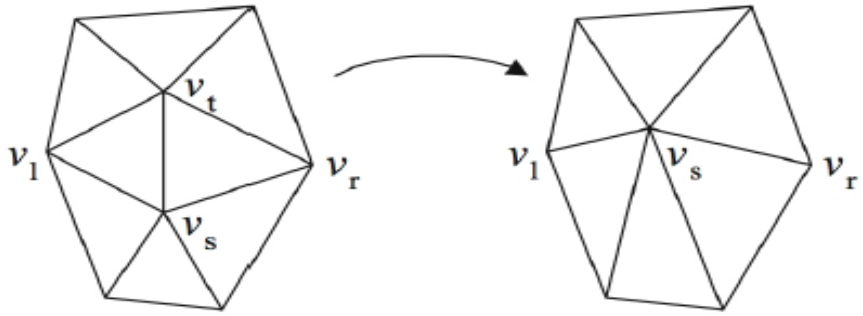
3.1.2. Kenar Kaynaştırma (Edge Contraction)

Kenar kaynaştırma temelli sadeleştirme, çokça kullanılan sadeleştirme algoritmalarından biridir. 1993 yılında Hoppe [3] tarafından önerilmiştir. Bir kenar alınır ve bu kenar bir köşe haline getirilir. Eski kenara bağlı tüm kenarlar tekrar düzenlenir. Şekil 3.4'de kenar kaynaştırma işleminin bir aşaması gösteriliyor.

Bu yöntemi kullanan birçok algoritma geliştirilmiştir. Bu yöntemler arasındaki temel fark, kaynaştırma için kenar seçmede uygulanan kriterdir. Kenar kaynaştırma işlemi, modelin topolojisinde değişiklikler üretebilir.

Ardışık kaynaştırma işlemleri ile modellerdeki boşluklar kapatılabilir ancak fakat birbirine bağlı olmayan bölgeler birleştirilemez. Bu işlem, modelin topolojisinde değişiklikler üretmesine rağmen, çok çözünürlüklü çizim için gerek duyulan bir özellik olmaktadır [41, 45].

Şekil 3.5'de Garland ve Heckbert'e ait çalışmada, kenar kaynaştırma ile elde edilen sadeleştirilmiş modeller gösteriliyor.



Şekil 3.4. Kenar kaynaştırma işlemi [3]



Şekil 3.5. Kademeli sadeleştirilen model [45]

3.2. Global Sadeleştirme Algoritmaları

Bu grup algoritmalar, mesh modelin tümü üzerinde işleyen algoritmaları ifade eder. Modeli tümü algoritmada girdi olarak kullanılır ve yeni bir model üretilir.

3.2.1. Poligon köşelerinin birleştirilmesi (vertex clustering)

Bu algoritma, orijinal modelin etrafına bir kutu yerleştirmekte ve bu kutu küçük parçalara bölünmektedir. Her bir hücrenin içinde yer alan poligon köşeleri tek bir poligon köşesine birleştirilmekte ve modelin poligon yüzleri de buna göre değiştirilmektedir [41, 42].

3.2.2. Örnekleme (sampling)

Bu yöntem, diğer yöntemlerden farklı olarak mevcut modeli basitleştirmek yerine, orijinal modelin yerini alabilecek yapıyı, yeni bir poligon köşeleri seti üzerinden yeniden oluşturmaktadır. Bu amaçla, poligon üzerinden, bir ızgaraya (grid) göre alınan poligon köşelerini kullanarak yeni bir model oluşturulmaktadır [41].

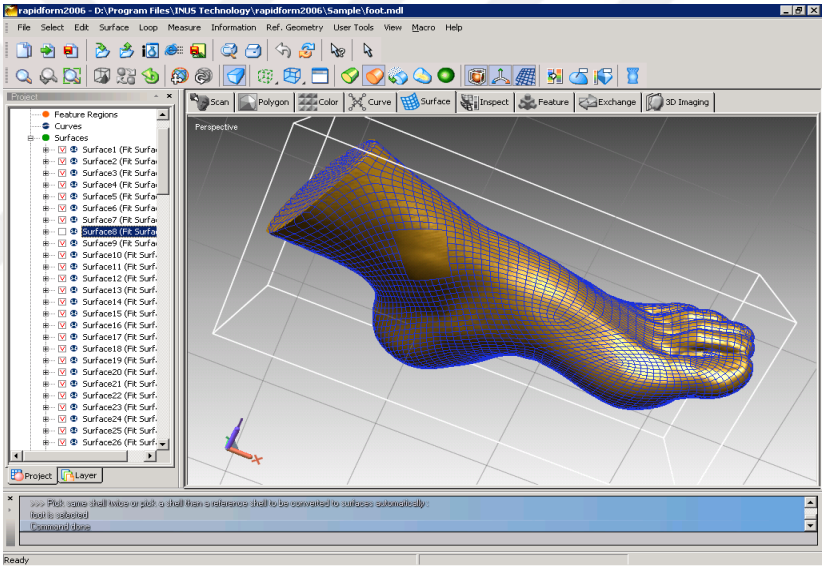
3.3. Dörtgensel Bölgelere Ayırma

Nokta bulutu verilerinden örme yüzey - mesh üretilmesi ile, noktaların, hangi düzlemlere ait olduğu belirgin hale gelir. Elde edilen poligon yüzeylerinin, anlamlı

bölgelere ayrılarak b-spline yama giydirme aşamasına hazırlanması gereklidir. Bu amaçla, çeşitli algoritmalara geliştirilmiştir. Bu algoritmalar, uygulama alanına göre çeşitlilik göstermektedir.

CAD uygulamalarında, keskin köşelerin ve mantıklı parçaların ayrıştırılması hedef iken, grafik ve animasyon endüstrisi, keskin köşelerin öncelikli olmadığı dörtgensel bölümlemeler üzerine yoğunlaşmıştır. Bazı uygulamalarda kullanıcı arayüzü aracılığı ile kullanıcıdan gelen bilgilere göre bölgeler ayrılmaktadır.

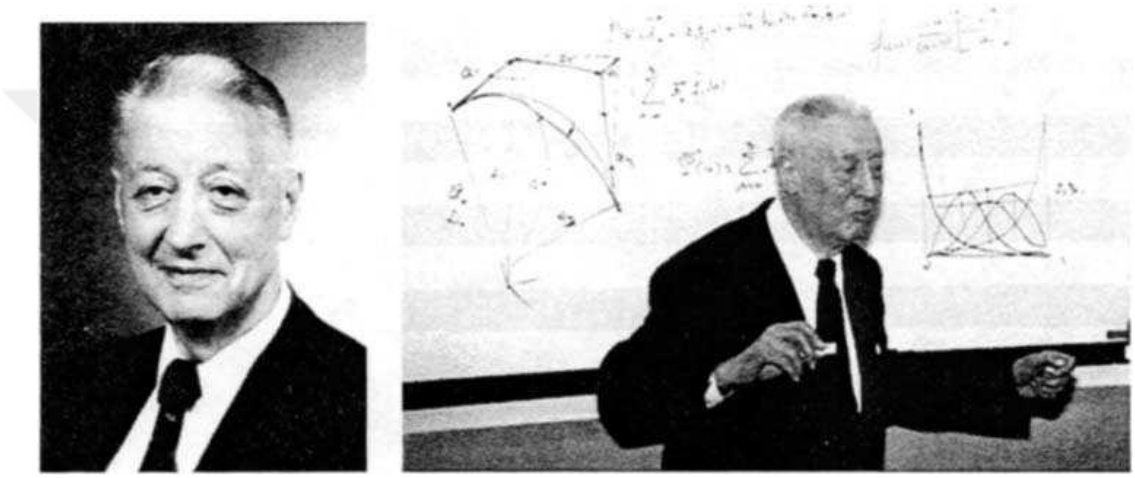
Bu tez kapsamında tek parçalı bir b-spline yama üretildiğinden, bölgelere ayırma aşaması ihmal edilmiştir. Şekil 3.6'da Rapidform uygulamasından alınan çok parçalı b-spline yamaların oluşturduğu tüm model gösterilmektedir.



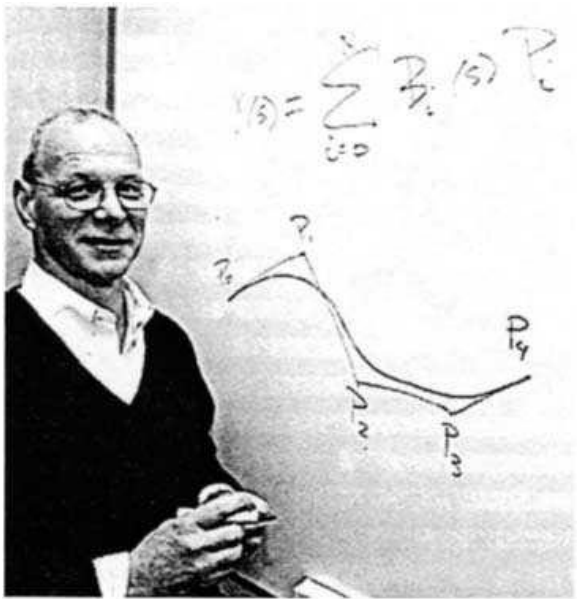
Şekil 3.6. RapidForm b-spline yama ekran görüntüsü

4. B-SPLİNE EĞRİ VE YÜZEYLER

Eğriler ve yüzeyler için B-spline teknikleri 1940'ların başında önerilmiş ve 1970'lerde Rich Riesenfeld tarafından geliştirilmiştir. B-spline yapısı, Pierre Bezier tarafıdan geliştirilmiş olan Bezier eğrilerinin genelleştirilmiş bir halidir. "B" harfi "basis" kelimesini ifade etmektedir [46].



Şekil 4.1. Pierre Bezier, fikirlerini açıklıyor [47]



Şekil 4.2. Rich Riesenfeld, 1970 [47]



Şekil 4.3. Boeing teknik ressam, fiziksel spline çizimi

B-spline eğrileri, Bezier eğrilerinde bulunan şu dezavantajları gidermek amacıyla geliştirilmiştir:

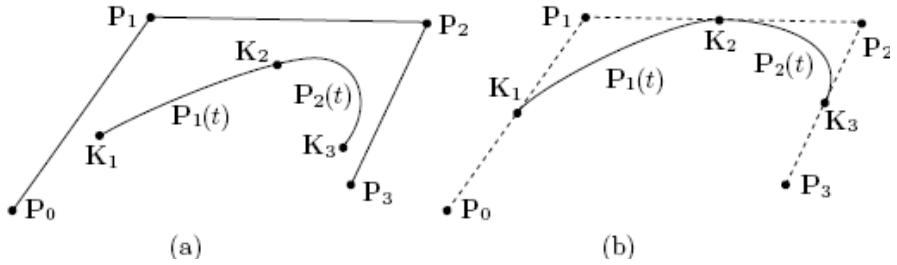
- Bezier eğrisinin derecesi, kontrol noktalarının sayısına bağlıdır
- Bezier eğrileri sadece global kontrol sağlar, lokal kontrol sağlamaz
- Bezier eğri parçalarının C^1 sürekliliğini sağlamak kolay iken, C^2 sürekliliğini sağlama konusunda yetersizdir.

B-spline eğrileri yerel kontrol sağlamaktadır. Şöyle ki, yerleri değiştirilen kontrol noktaları eğrinin sadece kontrol noktasından etkilenen komşuluklarını değiştirmektedir. Ayrıca B-spline eğriler ile, eğri parçalarının istenen derecede süreklilikleri sağlanabilmektedir [46]. Örneğin iki eğri parçası C^2 süreklilik ile birleştirildiğinde, eğrilerin birleştirme noktasındaki birinci ve ikinci türevlerinin eşit olması istenmektedir.

İki eğrinin birleşme noktasında C^n süreklilik istenmesi durumunda, her iki eğrinin de n dereceden polinomlar olması gerekir.

B-spline eğrileri, kontrol noktaları ile tanımlanır ve hedeflenen bir nokta kümesine veya yüzeye yaklaşım yapmak için kullanılır. Ayrıca noktaların bulunduğu uzaydan bağımsız olarak tanımlanan düğüm vektörü olarak ifade edilen bir parametre uzayına ihtiyaç bulunur. Düğüm vektörü düzgün dağılımlı – uniform olabileceği gibi, hedef noktaları daha iyi yaklaşım yapmak üzere çeşitli metotlarla düzgün olmayan bir

dağılım – nonuniform şeklinde de tanımlanabilir. Nokta kümesi enterpolasyonu ve yaklaşımı amaçlı uygulamalarda, düğüm vektörü hesaplama metotları sonraki aşamalarda anlatılmaktadır. Şekil 4.4'de ikinci derece uniform b-spline oluşum yapısı gösterilmektedir.



Şekil 4.4. İkinci derece b-spline gösterimi [46]

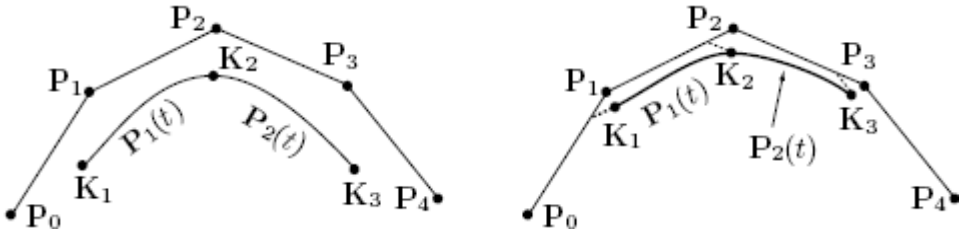
B-spline iki adet eğri parçasından oluşmaktadır. $P_i(t)$ eğri parçası, P_{i-1}, P_i ve P_{i+1} kontrol noktalarına göre oluşur. Denklem (4.1)'de bu iki eğri parçasının ilk seviye denklemi gösteriliyor. Burada M, $3x3$ boyutlarında baz matrisi ifade ediyor ve eğrilerin birleşim noktasındaki birinci derece türev sürekliliği gereksinimine göre hesaplanır. Baz matrisin, öz yinelemeli olarak, bilgisayarlaştırılmış hesaplama metodu sonraki aşamalarda anlatılıyor.

$$P_i(t) = (t^2, t, 1) M \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}, \; i = 1, 2 \tag{4.1}$$

$$P_i(t) = \frac{1}{2}(t^2, t, 1) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}, \tag{4.2}$$

$$= \frac{1}{2}(t^2 - 2t + 1) P_{i-1} + \frac{1}{2}(-2t^2 + 2t + 1) P_i + \frac{t^2}{2} P_{i+1}, i = 1, 2.$$

Üçüncü dereceden uniform B-spline yapılanması ise Şekil 4.5'de gösteriliyor. Denklem (4.3)'de kübik uniform b-spline yapısı için ilk denklem gösteriliyor. Bu denklemde M, $4x4$ boyutlarında baz matrisi ifade eder. Üçüncü derece spline segmentlerinin birleşme noktasında, C^2 türev sürekliliği şartı sağlanmalıdır.



Şekil 4.5. Kübik uniform b-spline yapılanması [46]

$$P_i(t) = (t^3, t^2, t, 1) M \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix} \tag{4.3}$$

Bu şartları sağlayan, uniform düğüm vektörüne sahip b-spline yapısında, M matrisi aşağıdaki şekilde hesaplanır: [46]

$$\begin{aligned} P_i(t) &= \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}, \\ &= \frac{1}{6}(-t^3 + 3t^2 - 3t + 1) P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4) P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1) P_{i+1} + \frac{t^3}{6} P_{i+2} \end{aligned} \tag{4.4}$$

Daha yüksek derecelerdeki uniform b-spline yapılandırmaları şu şekilde özetlenebilir [46]:

$$P_i(t) = (t^3, t^2, t, 1) M \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ \cdot \\ \cdot \\ P_{i+n-1} \end{pmatrix} \tag{4.5}$$

M matrisini elemanları ise şu şekilde ifade edilir:

$$m_{ij} = \frac{1}{n!} \binom{n}{i} \sum_{k=j}^{n} (n-k)^i (-1)^{k-j} \binom{n+1}{k-j} \tag{4.6}$$

Denklem (4.7)'de uniform düğüm vektörüne sahip çeşitli derecelerdeki b-spline yapısı için geçerli baz matrisler gösteriliyor.

$$M_1 = \frac{1}{1!}\begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$M_2 = \frac{1}{2!}\begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

$$M3 = \frac{1}{3!}\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \tag{4.7}$$

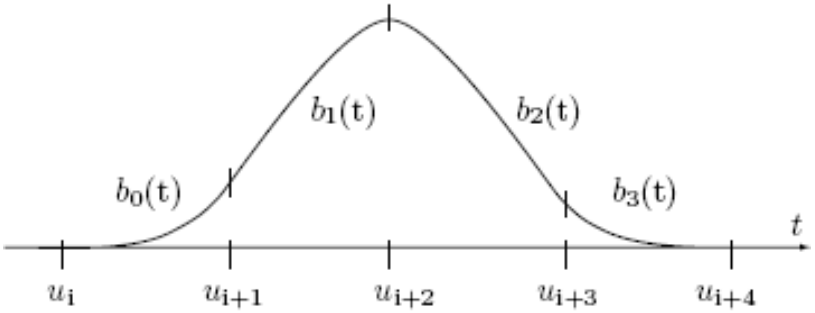
$$M_4 = \frac{1}{4!}\begin{pmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -6 & -6 & 6 & 0 \\ -4 & -12 & 12 & 4 & 0 \\ 1 & 11 & 11 & 1 & 0 \end{pmatrix}$$

4.1. B-Spline Düğüm Vektör Yaklaşımı

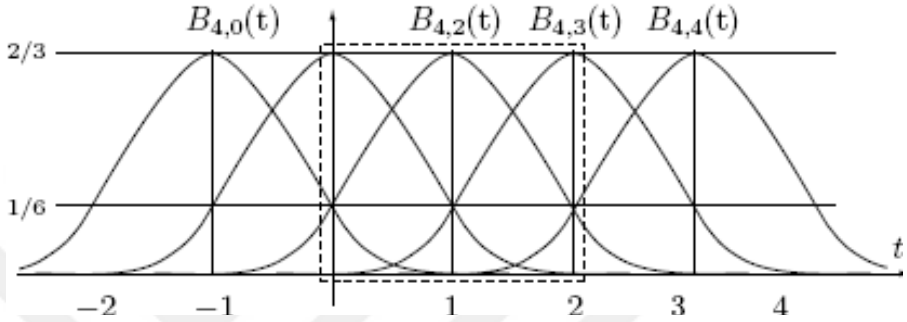
Düğüm vektörü ile gösterimde eğri, kontrol noktalarının, ağırlık fonksiyonlarına göre, ağırlıklı ortalamaları şeklinde ifade edilir [46].

$$P_t = \sum_{i=0}^{n} P_i B_{n,i}(t) \tag{4.8}$$

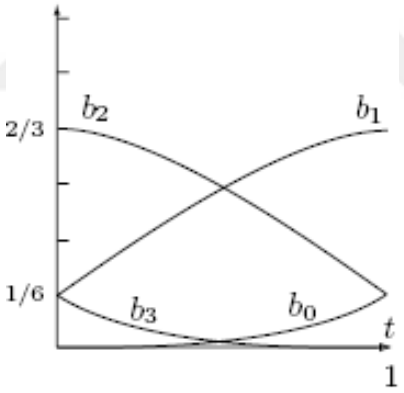
Şekil 4.7'de, beş adet kontrol noktası ile tanımlanan, iki adet kübik b-spline segmentinden oluşan b-spline eğrisi için, $B_{4,0}(t) .. B_{4,4}(t)$ ağırlık fonksiyonlarının oluşumu gösteriliyor. Her bir b-spline segmenti, t parametresi, $u_i .. u_{i+1}$ aralığında değişirken, bu segmente etki eden kontrol noktalarının, ağırlık fonksiyonları ile ağırlıklandırılmış ortalamaları kullanılarak hesaplanır. Örnekte, t parametresi $[0,2]$ aralığında değişmektedir. Ağırlık fonksiyonları, şekilde 4.6'da gösterilen, $b_0(t)$, $b_1(t)$, $b_2(t)$, $b_3(t)$ isimli dört parçadan oluşan bir çan eğrisi şeklindedir. Dört parçanın her biri birer kübik fonksiyon olarak elde edilmelidir.



Şekil 4.6. Çan eğrisi, dört parçalı olarak tanımlanıyor [46]



Şekil 4.7. Ağırlık fonksiyonları, dört parçalı çan eğrilerinden oluşur [46]



Şekil 4.8. 0-1 Aralığında ağırlık fonksiyonları

$b_i(t)$ fonksiyonlarının şu özellikleri sağlaması gereklidir:

–Fonksiyonların herhangi bir düğüm vektörü aralığında değer toplamlarının 1 olması (barycentric, kütle merkezi)

– $b_0(1) = b_1(0)$, $b_1(1) = b_2(0)$, $b_2(1) = b_3(0)$ birleşme noktalarında C^2 türev sürekliliğinin sağlanması

– $b_0(t)$ ve bu fonksiyonun ilk iki türevi, $b_0(0)$ noktasında sıfır olmalıdır

– $b_3(t)$ ve bu fonksiyonun ilk iki türevi, $b_3(1)$ noktasında sıfır olmalıdır

Gerekli hesaplamalar yapıldığında, çan eğrisinin parça fonksiyonları Denklem (4.9)'daki gibi elde edilir [46].

$$b_0(t) = \frac{1}{6}t^3$$

$$b_1(t) = \frac{1}{6}(1 + 3t + 3t^2 - 3t^3)$$

$$b_2(t) = \frac{1}{6}(4 - 6t^2 + 3t^3) \tag{4.9}$$

$$b3(t) = \frac{1}{6}(1 - 3t + 3t^2 - t^3)$$

Bu fonksiyonların elde edilmesinden sonra, iki adet kübik b-spline segmentinden oluşan eğri, Denklem (4.10)'da gösterildiği gibi, kontrol noktalarının ağırlıklı ortalamaları şeklinde ifade edilir.

$$P_t = \sum_{i=0}^{n} P_i B_{4,i}(t) \tag{4.10}$$

4.2. B-Spline Baz Fonksiyonlarının Özyinelemeli Hesaplanması

Bilgisayar ortamında b-spline eğri hesaplamalarında, düğüm vektörü girdisine göre özyinelemeli olarak işleyen Cox-de Boor algoritması kullanılır.

p'inci dereceden bir b-spline eğrisi şu şekilde tanımlanmış olsun:

$$C_{(u)} = \sum_{i=0}^{n} N_{i,p}(u) P_i \tag{4.11}$$

Bu tanımlamada, P_i kontrol noktalarını ifade ederken, $N_{i,p}(u)$ değerleri p'inci derece B-spline baz fonksiyonlarını ifade eder. U vektörü, nesne uzayından bağımsız olarak işleyen, sürekli artan reel sayılardan oluşur ve düğüm vektörü olarak isimlendirilir.

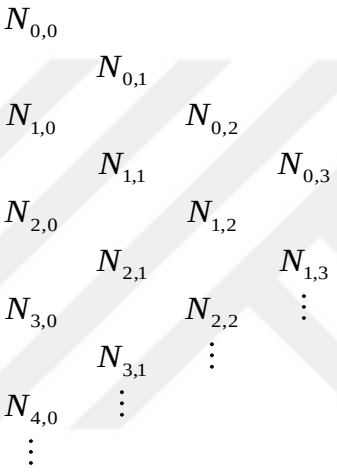
U vektörünün her bir u_i elemanı düğüm olarak adlandırılır. Bu tanımlamalar altında, p'inci dereceden b-spline'ın i'inci baz fonksiyonu, Cox-de Boor algoritması ile bilgisayar ortamında öz yinelemeli olarak hesaplanır.

Cox-de Boor algoritması Denklem (4.12)'de verilmiştir.

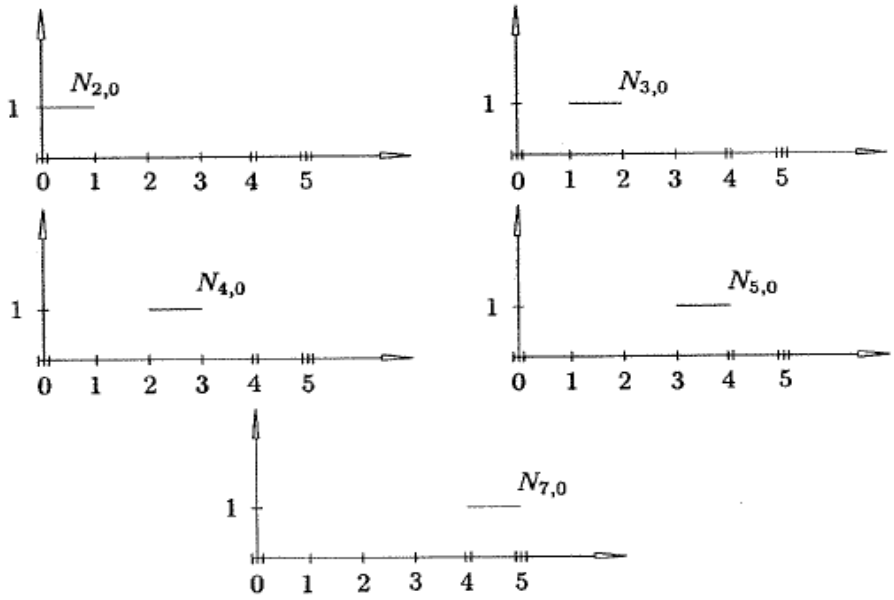
$$N_{i,0}(u) = \begin{cases} 1, & u_i \le u \le u_{i+1} \\ 0, & diğer \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{4.12}$$

Şekil 4.9'da p'inci derece baz fonksiyonları hesaplarken ara kademelerde oluşan üçgen tablo gösteriliyor.

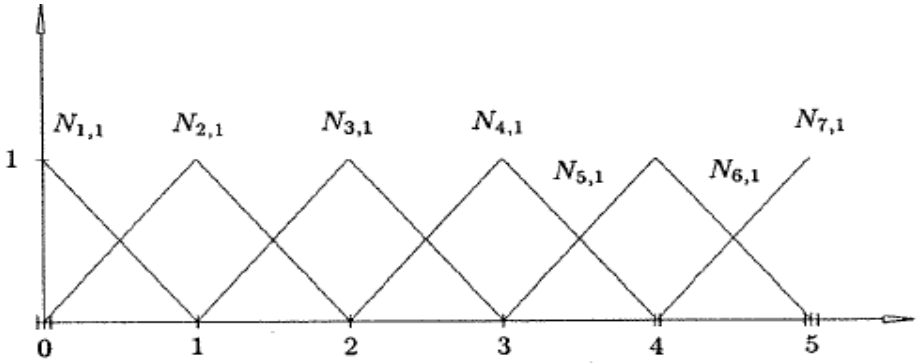


Şekil 4.9. p'inci derece baz fonksiyonları tablosu

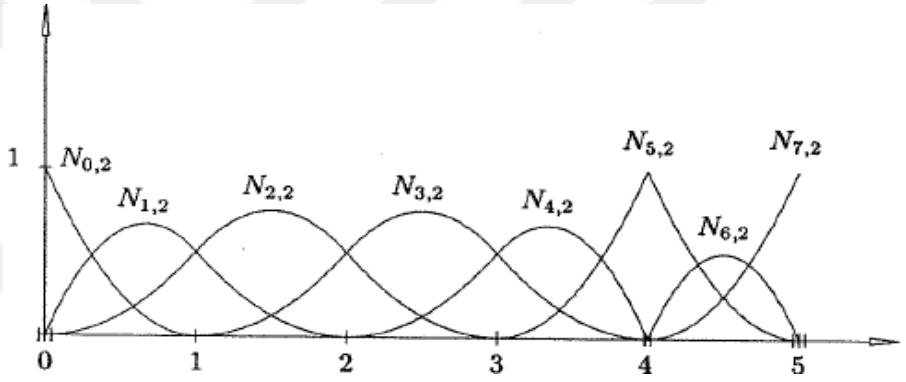


Şekil 4.10. Sıfır derece baz fonksiyonları U={0,0,0,1,2,3,4,4,5,5,5}

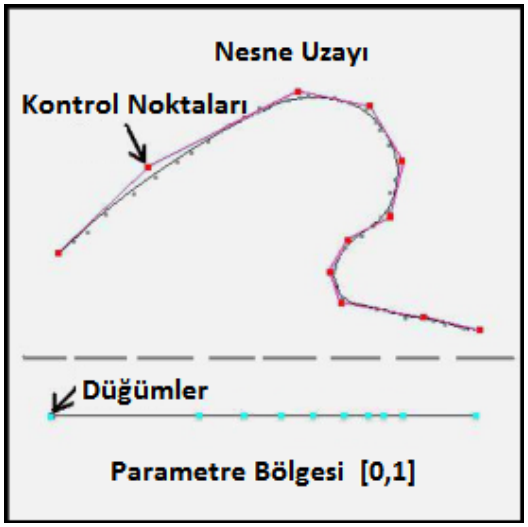
Şekil 4.10, 4.11 ve 4.12'de ikinci dereceden polinomlar ile bina edilen b-spline için U={0,0,0,1,2,3,4,4,5,5,5} düğüm vektörüne göre üretilen baz fonksiyonlarının üretim aşamaları gösteriliyor [48].



Şekil 4.11. Birinci derece baz fonksiyonları U={0,0,0,1,2,3,4,4,5,5,5}



Şekil 4.12. İkinci derece baz fonksiyonları U={0,0,0,1,2,3,4,4,5,5,5}



Şekil 4.13. Parametre uzayı ve nesne uzayının bağımsızlığı

Şekil 4.13'de düğüm vektörünü uzayı ve nesne uzayının bağımsız durumu gösteriliyor. Düğüm noktalarının seçimi, b-spline eğrisi üzerinde önemli etki göstermektedir. Düğüm vektörü, monoton artan reel sayılardan oluşmalıdır. Düğüm vektörünün elemanları doğrusal artan elemanlardan oluşuyor ise, uniform – monoton artan düğüm vektörü olarak anılır.

$\{0 \quad 1 \quad 2 \quad 3 \quad 4\}$
$\{-0{,}2 \quad -0{,}1 \quad 0 \quad 0{,}1 \quad 0{,}2\}$

Pratik kullanımda, uniform düğüm vektörü 0 ile başlar ve 1'er adımlarla ilerleyen tamsayılar seçilir. Veya adım artışları 0-1 aralığına normalize edilir.

$\{0 \quad 0{,}25 \quad 0{,}5 \quad 0{,}75 \quad 1{,}0\}$

p'inci dereceden baz fonksiyonları bina edecek olan uniform düğüm vektörünün başlangıç ve sonundaki değer (p+1) kere tekrar edilir.

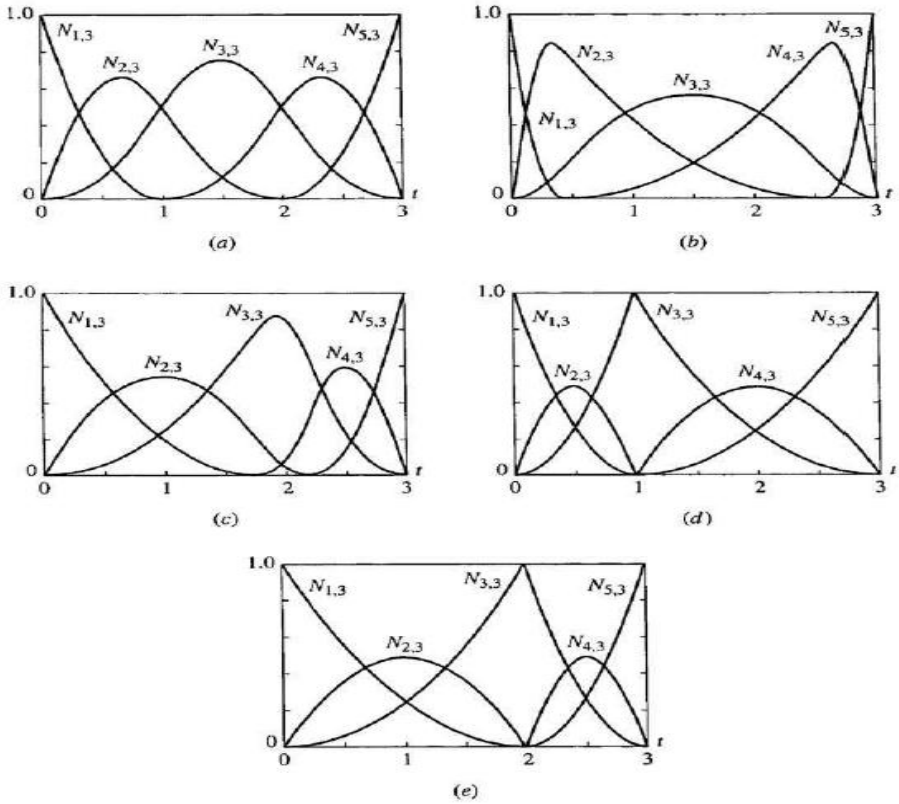
$p = 1\{0012344\}$
$p = 2\{00012333\}$
$p = 3\{000012222\}$

Veya normalize olarak ifade edilirse:

$p = 1\{001/41/23/411\}$
$p = 2\{0001/32/3111\}$
$p = 3\{00001/21111\}$

Vektörün elemanları düzensiz olarak artıyor ise, vektör non-uniform düzensiz artan düğüm vektörü olarak isimlendirilir.

B-spline eğri ve yüzey enterpolasyon problemlerinde, hedef noktaların arasındaki mesafeler ve açılar kullanılarak, gerekli parametre adımlarının ve düğüm vektörünün hesaplanması gerekir. Bu aşama, parametrelendirme olarak anılır.



Şekil 4.14. Düzensiz artımlı – nonuniform düğüm vektörleri ile 2. derece baz fonksiyonlar [47] (a). $U = \{0\ 0\ 0\ 1\ 2\ 3\ 3\ 3\}$ (b). $U = \{0\ 0\ 0\ 0{,}4\ 2{,}6\ 3\ 3\ 3\}$ (c). $U = \{0\ 0\ 0\ 1{,}8\ 2{,}2\ 3\ 3\ 3\}$ (d). $U = \{0\ 0\ 0\ 1\ 1\ 3\ 3\ 3\}$ (e). $U = \{0\ 0\ 0\ 2\ 2\ 3\ 3\ 3\}$

4.3. B-spline Eğri Anlık Değer Hesabı

İkinci dereceden b-spline kurgusu için $U = \{0\ 0\ 0\ 1\ 2\ 3\ 4\ 4\ 5\ 5\ 5\}$ düğüm vektörü verilmiş olsun. $u = 5/2$ düğüm değerinde, b-spline eğrisinin değeri hesaplanmak istendiğinde şu adımlar takip edilir.

Verilen anlık düğüm değeri $u = 5/2$, $u \in [u_4 u_5)$ aralığında olduğundan span/sıra indeks değeri $i = 4$ olur. Bu durumda, şu baz fonksiyon değerlerinin hesaplanması gerekir:

$$N_{2,2(5/2)}$$
$$N_{3,1(5/2)}$$
$$N_{4,0(5/2)} \quad N_{3,2(5/2)}$$
$$N_{4,1(5/2)}$$
$$N_{4,2(5/2)}$$

u=5/2 değeri Denklem (4.12)'de, verilen düğüm vektörü girdisi ile hesaplandığında,

$$N_{4,0(5/2)} = 1$$
$$N_{3,1(5/2)} = 1/2 \quad N_{4,1(5/2)} = 1/2$$
$$N_{2,2(5/2)} = 1/8 \quad N_{3,2(5/2)} = 6/8 \quad N_{4,2(5/2)} = 1/8$$

değerleri bulunur. Denklem (4.11)'deki hesaplama yapılır ise, $N_{2,2(5/2)} = 1/8$, $N_{3,2(5/2)} = 6/8$, $N_{4,2(5/2)} = 1/8$ değerleri, ilgili kontrol noktalarının ağırlıkları olarak işler. Eğrinin $u = 5/2$ düğüm değerindeki anlık değerli, $C(5/2)$ şu şekilde bulunur [48]:

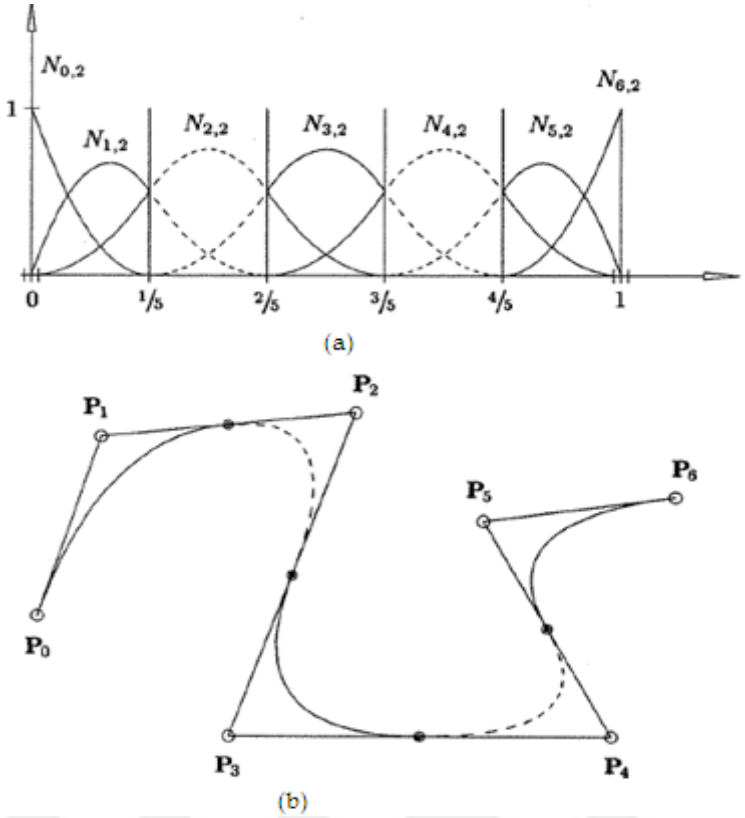
$$C\left(\frac{5}{2}\right) = \frac{1}{8}P_2 + \frac{6}{8}P_3 + \frac{1}{8}P_4$$

Şekil 4.15'de bu örnekteki düğüm vektörünün normalize edilmiş durumuna göre, ikinci derece baz fonksiyonları ve kontrol noktalarına göre oluşan ikinci derece b-spline eğri parçaları gösteriliyor.

Uygulamada Piegl'in kitabındaki [48] algoritmaların matlab versiyonları kullanılmıştır. Bu kitabın kodları, geometri kernelleri (ACIS, GRANITE, OPENCASCADE v.b.) seviyesinde ticari olarak CAD-CAM geliştirici firmalara sunulmaktadır.

4.4. B-Spline Yüzey Tanımı

B-spline yüzey, iki adet b-spline eğrinin kartezyen çarpımı şeklinde elde edilmektedir. Bir kübik b-spline segmentine dört adet kontrol noktası etki eder iken, bir bi-kübik b-spline yüzey parçasına ilgili komşuluk alanında bulunan, kontrol-net olarak isimlendirilen 16 adet kontrol noktası etki eder. Şekil 4.16'da bir b-spline yüzey segmenti için kontrol noktaları, kontrol ağı ve b-spline yüzeyi gösteriliyor. Denklem (4.13)'de b-spline yüzey tanımlaması gösteriliyor.

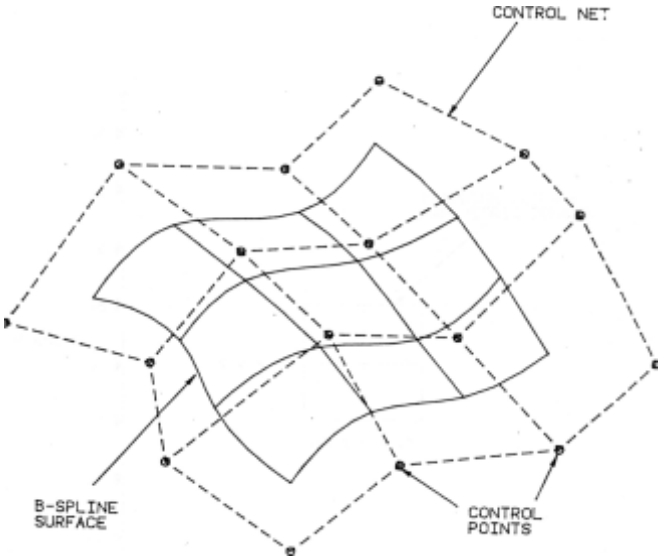


Şekil 4.15. (a). İkinci derece baz fonksiyonları (b). b-spline eğri [48]

$$S_{(u,v)} = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)P_{ij} \tag{4.13}$$

U ve V düğüm vektörleri Denklem (4.14)'deki gibi tanımlanır. U vektörü $r+1$ adet düğüme, V vektörü ise, $s+1$ adet düğüme sahiptir. U vektörünün başındaki 0 ve sonundaki 1 değerleri $p+1$ adet tekrar eder. V vektörü için de benzer şekilde, $q+1$ adet tekrar yapılır. B-spline terminolojisinde bu durum "clamping" olarak isimlendirilir. Bu şekilde üretilen eğrinin başlangıç ve son noktası, ilk ve son kontrol noktasına erişmeye zorlanır.

$$\begin{aligned} U &= \{0,\ldots,0,u_{p+1},\ldots,u_{r-p-1},1,\ldots,1\} \\ V &= \{0,\ldots,0,v_{q+1},\ldots,v_{s-q-1},1,\ldots,1\} \\ r &= n+p+1 \\ s &= m+q+1 \end{aligned} \tag{4.14}$$



Şekil 4.16. Bi-kübik b-spline yüzey segmenti, etki eden 16 kontrol noktası

Örnek olarak, $p = q = 2$ olan, ikinci derece b-spline yüzeyin denklemi şu şekilde verilmiş olsun:

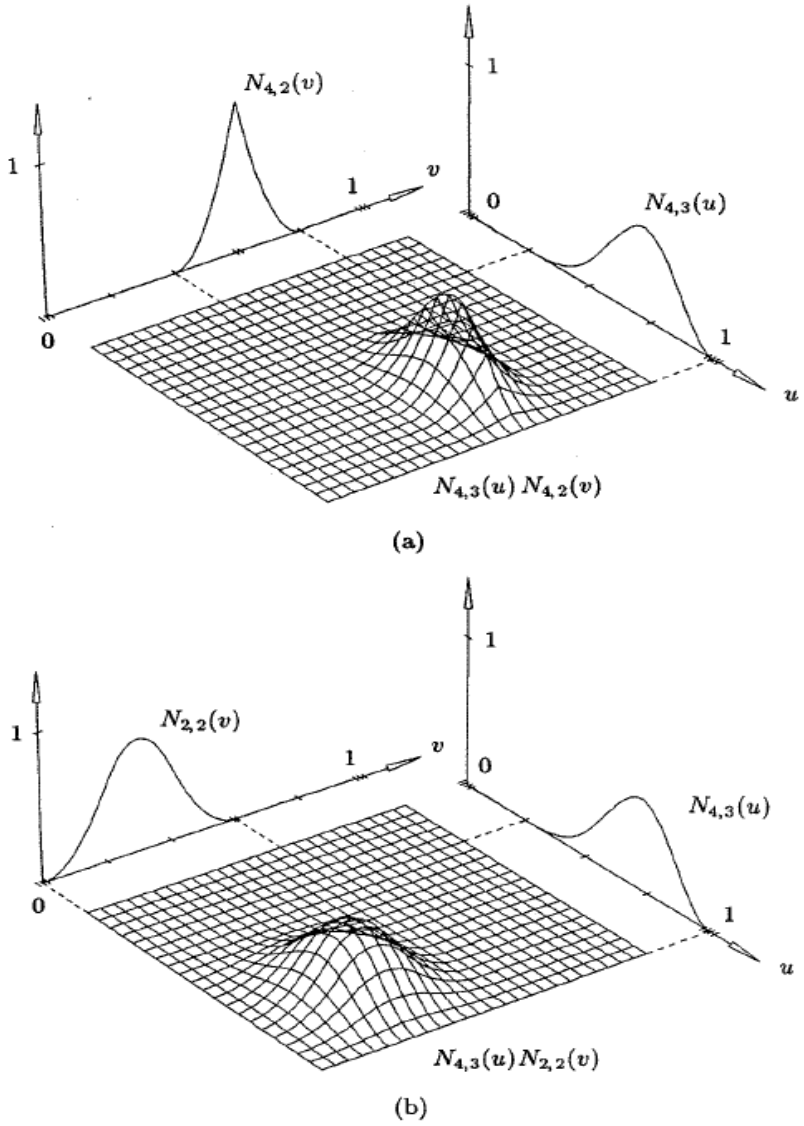
$$S_{(u,v)} = \sum_{i=0}^{4}\sum_{j=0}^{5} N_{i,2}(u) M_{j,2}(v) P_{ij}$$

$$U = \{0,0,0,\frac{2}{5},\frac{3}{5},1,1,1\} \qquad V = \{0,0,0,\frac{1}{5},\frac{1}{2},\frac{4}{5},1,1,1\}$$

Yüzeyin $S\left(\frac{1}{5},\frac{3}{5}\right)$ düğüm noktasında anlık değerini hesaplama için, önce verilen u ve v değerlerinin aralığı bulunur. $\frac{1}{5} \in [u_2, u_3)$ ve $\frac{3}{5} \in [v_4, v_5)$ olduğundan, bu düğüm aralıklarında etkin olan baz fonksiyonların değerleri hesaplanır. $S\left(\frac{1}{5},\frac{3}{5}\right)$ yüzey anlık değeri ise şu şekilde hesaplanır [48]:

$$S\left(\frac{1}{5},\frac{3}{5}\right) = \begin{bmatrix} N_{0,2}\left(\frac{1}{5}\right) & N_{1,2}\left(\frac{1}{5}\right) & N_{2,2}\left(\frac{1}{5}\right) \end{bmatrix} \times \begin{bmatrix} P_{0,2} & P_{0,3} & P_{0,4} \\ P_{1,2} & P_{1,3} & P_{1,4} \\ P_{2,2} & P_{2,3} & P_{2,4} \end{bmatrix} \times \begin{bmatrix} M_{2,2}\left(\frac{3}{5}\right) \\ M_{3,2}\left(\frac{3}{5}\right) \\ M_{4,2}\left(\frac{3}{5}\right) \end{bmatrix}$$

Şekil 4.17'de $U = \{0\ 0\ 0\ 0\ \frac{1}{4}\ \frac{1}{2}\ \frac{3}{4}\ 1\ 1\ 1\ 1\}$ $V = \{0\ 0\ 0\ \frac{1}{5}\ \frac{2}{5}\ \frac{3}{5}\ \frac{3}{5}\ \frac{4}{5}\ 1\ 1\ 1\}$ düğüm vektörlerine göre işleyen, kübik ve ikinci derece baz fonksiyonların kartezyen çarpımı gösteriliyor. Kübik baz fonksiyonun aralığı sabit kalır iken, ikinci derece fonksiyonun indisi değiştiriliyor.



Şekil 4.17. Kartezyen çarpım gösterimi (a). $N_{4,3}(u)N_{4,2}(v)$ (b). $N_{4,3}(u)N_{2,2}(v)$) [48]

4.5. B-spline Eğri Enterpolasyonu

Bezier ve b-spline eğrileri, bilgisayar grafikleri, animasyon uygulamalarında ve CAD-CAM yüzey modelleme aşamalarında, tasarımcıdan gelen kontrol noktaları ve düğüm vektörü verilerine göre ileri yönde çalışmaktadır. Ancak tersine mühendislik

uygulamalarında ve veri enterpolasyonu amaçlı kullanımlarda bu işlemin tersten yürütülmesi gerekir. Bu durumda eğrinin veya yüzeyin enterpole edeceği nokta verileri sağlanır iken, kontrol noktaları ve gerekli düğüm vektörünün üretilmesi istenir. Bu aşamada, işlemler tersine yürütülür.

B-spline eğrinin denklemi şu şekilde verilmiş olsun:

$$C_{(u)} = \sum_{i=0}^{n} N_{i,p}(u) P_i \tag{4.15}$$

Parametrelendirme neticesinde, düğüm vektörü üzerinde her bir parametre değerine karşılık hesaplanacak olan noktanın değeri ise şu şekilde ifade edilir:

$$D_k = C\left(t_k\right) = \sum_{i=0}^{n} N_{i,p}(t_k) P_i, \quad 0 \le k \le n \tag{4.16}$$

Parametre değerlerindeki baz fonksiyonların değerleri N matrisine yazılır:

$$N = \begin{bmatrix} N_{0,p}(t_0) & N_{1,p}(t_0) & N_{2,p}(t_0) & \cdots & N_{n,p}(t_0) \\ N_{0,p}(t_1) & N_{1,p}(t_1) & N_{2,p}(t_1) & \cdots & N_{n,p}(t_1) \\ \vdots & & & \ddots & \vdots \\ N_{0,p}(t_n) & N_{1,p}(t_n) & N_{2,p}(t_n) & \cdots & N_{n,p}(t_n) \end{bmatrix} \tag{4.17}$$

Hesaplanan D hedef noktaları ve P kontrol noktaları matris formunda şu şekilde ifade edilir:

$$D = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix}, \quad P = \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_n \end{bmatrix} \tag{4.18}$$

Kontrol noktaları verilmiş iken, ileri yönde hesaplamada, b-spline eğri denklemi matris formunda şu şekilde ifade edilir:

$$D=N.P \tag{4.19}$$

Kontrol noktalarının hesaplanacağı eğri enterpolasyonu işlemlerinde ise P kontrol noktalarının elde edilmesi için aşağıdaki aşamalar uygulanır:

$$D = N.P$$

$$N^T D = N^T NP \tag{4.20}$$

$$P = \left(N^T N\right)^{-1}\left(N^T D\right)$$

Kübik eğri enterpolasyonu için örnek olarak, D noktaları şu şekilde verilmiş olsun:

$$D_k = \{(0,0),(3,4),(-1,4),(-4,0),(-4,-3)\}$$

4.9.2'de anlatılan kiriş mesafesine (chord) göre parametrelendirme yapılır ise, öncelikle hedef noktalar arasındaki mesafeler bulunur:

$$|D_1 - D_o| = 5,\ |D_2 - D_1| = 4,\ |D_3 - D_2| = 5,\ |D_4 - D_3| = 3$$

Mesafeler toplamı 17 olduğundan parametre değerleri şu şekilde hesaplanır:

$$t_0 = 0$$

$$t_1 = \frac{5}{17}$$

$$t_2 = \frac{9}{17}$$

$$t_3 = \frac{14}{17}$$

$$t_4 = 1$$

Singular-tekil matris durumundan sakınmak için, Denklem (4.38)'de anlatılan ortalama-averaging tekniği kullanılarak, parametre değerlerinden, clamping-sıkma içeren düğüm vektörü hesaplanır.

$$u_4 = \frac{1}{3}\left(\frac{5}{17} + \frac{9}{17} + \frac{14}{17}\right) = \frac{28}{51},$$

$$U = \left\{0\ 0\ 0\ 0\ \frac{28}{51}\ 1\ 1\ 1\ 1\right\}$$

Düğüm vektörü girdisine göre ilgili baz fonksiyonları hesaplanarak aşağıdaki lineer denklem sistemine doldurulur:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ N_{0,3}\left(\frac{5}{17}\right) & N_{1,3}\left(\frac{5}{17}\right) & N_{2,3}\left(\frac{5}{17}\right) & N_{3,3}\left(\frac{5}{17}\right) & 0 \\ N_{0,3}\left(\frac{9}{17}\right) & N_{1,3}\left(\frac{9}{17}\right) & N_{2,3}\left(\frac{9}{17}\right) & N_{3,3}\left(\frac{9}{17}\right) & 0 \\ 0 & N_{1,3}\left(\frac{14}{17}\right) & N_{2,3}\left(\frac{14}{17}\right) & N_{3,3}\left(\frac{14}{17}\right) & N_{4,3}\left(\frac{14}{17}\right) \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

D hedef noktaları enterpole edilmesi istenen girdi noktalar olarak verildiğinden, P kontrol noktaları Denklem (4.20)'de gösterilen matris işlemleri ile elde edilir. Piegl ve Tiller'in kitabında [48], bu işlemlerin bilgisayarlaştırılması aşaması ham algoritma olarak gösteriliyor.

4.6. B-spline Yüzey Enterpolasyonu

S yüzeyi, p'inci ve q'uncu derece b-spline Kartezyen çarpımı olarak şu şekilde tanımlanmış olsun:

$$S(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(u) N_{j,q}(v) P_{i,j} \tag{4.21}$$

u ve v düğüm vektörlerinin parametrelendirildiği kesme yerlerinde, c ve d harfleri ile indisli, $u = s_c$ ve $v = t_d$ değerleri yer alsın. Bu durumda yüzey formülü şu şekilde ifade edilir:

$$D_{cd} = S(s_c, t_d) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(s_c) N_{j,q}(t_d) P_{i,j} \tag{4.22}$$

$N_{i,p}(s_c)$, j indeksinden bağımsız olduğundan, dışarı alınabilir. Bu durumda, b-spline yüzey enterpolasyonu, bir dizi b-spline eğri enterpolasyonu olarak ifade elde edilir:

$$D_{cd} = S(s_c, t_d) = \sum_{i=0}^{m} N_{i,p}(s_c) \left(\sum_{j=0}^{n} N_{j,q}(t_d) P_{i,j} \right) \tag{4.23}$$

Matris formunda tanımlama yapmak üzere b-spline kartezyen çarpım yüzey ifadesi şu şekilde verilmiş olsun:

$$S(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}Z_{i,p}(u)N_{j,q}(v)P_{i,j} \tag{4.24}$$

Bu yüzeyin matris formundaki gösterimi için, C kontrol noktaları ve D hedef noktaları şu şekilde ifade edilir:

$$C=\begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,n} \\ c_{1,0} & c_{1,1} & & c_{1,n} \\ \vdots & & \ddots & \vdots \\ c_{m,0} & c_{m,1} & \cdots & c_{m,n} \end{bmatrix}$$
$$D=\begin{bmatrix} d_{0,0} & d_{0,1} & \cdots & d_{0,n} \\ d_{1,0} & d_{1,1} & & d_{1,n} \\ \vdots & & \ddots & \vdots \\ d_{m,0} & d_{m,1} & \cdots & d_{m,n} \end{bmatrix} \tag{4.25}$$

Benzer şekilde, ilgili Z ve N baz fonksiyonları şu şekilde ifade edilir:

$$Z=\begin{bmatrix} Z_{0,p}(s_0) & Z_{1,p}(s_0) & Z_{2,p}(s_0) & \cdots & Z_{m,p}(s_0) \\ Z_{0,p}(s_1) & Z_{1,p}(s_1) & Z_{2,p}(s_1) & \cdots & Z_{m,p}(s_1) \\ \vdots & & & \ddots & \vdots \\ Z_{0,p}(s_m) & Z_{1,p}(s_m) & Z_{2,p}(s_m) & \cdots & Z_{m,p}(s_m) \end{bmatrix}$$
$$N=\begin{bmatrix} N_{0,q}(t_0) & N_{1,q}(t_0) & N_{2,q}(t_0) & \cdots & N_{n,q}(t_0) \\ N_{0,q}(t_1) & N_{1,q}(t_1) & N_{2,q}(t_1) & \cdots & N_{n,q}(t_1) \\ \vdots & & & \ddots & \vdots \\ N_{0,q}(t_n) & N_{1,q}(t_n) & N_{2,q}(t_n) & \cdots & N_{n,q}(t_n) \end{bmatrix} \tag{4.26}$$

Bu durumda, b-spline yüzey matris formunda Denklem (4.27) ile ifade edilir. Hesaplanması istenen C kontrol noktalarına ise Denklem (4.28) kullanılarak elde edilir [49]:

$$\begin{bmatrix} d_{0,0} & d_{0,1} & \cdots & d_{0,n} \\ d_{1,0} & d_{1,1} & & d_{1,n} \\ \vdots & & \ddots & \vdots \\ d_{m,0} & d_{m,1} & \cdots & d_{m,n} \end{bmatrix} =$$

$$\begin{bmatrix} Z_{0,p}(s_0) & Z_{1,p}(s_0) & Z_{2,p}(s_0) & \cdots & Z_{m,p}(s_0) \\ Z_{0,p}(s_1) & Z_{1,p}(s_1) & Z_{2,p}(s_1) & \cdots & Z_{m,p}(s_1) \\ \vdots & & & \ddots & \vdots \\ Z_{0,p}(s_m) & Z_{1,p}(s_m) & Z_{2,p}(s_m) & \cdots & Z_{m,p}(s_m) \end{bmatrix} \times \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,n} \\ c_{1,0} & c_{1,1} & & c_{1,n} \\ \vdots & & \ddots & \vdots \\ c_{m,0} & c_{m,1} & \cdots & c_{m,n} \end{bmatrix} \tag{4.27}$$

$$\times \begin{bmatrix} N_{0,q}(t_0) & N_{1,q}(t_0) & N_{2,q}(t_0) & \cdots & N_{n,q}(t_0) \\ N_{0,q}(t_1) & N_{1,q}(t_1) & N_{2,q}(t_1) & \cdots & N_{n,q}(t_1) \\ \vdots & & & \ddots & \vdots \\ N_{0,q}(t_n) & N_{1,q}(t_n) & N_{2,q}(t_n) & \cdots & N_{n,q}(t_n) \end{bmatrix}$$

$$D = Z\,C\,N^T$$

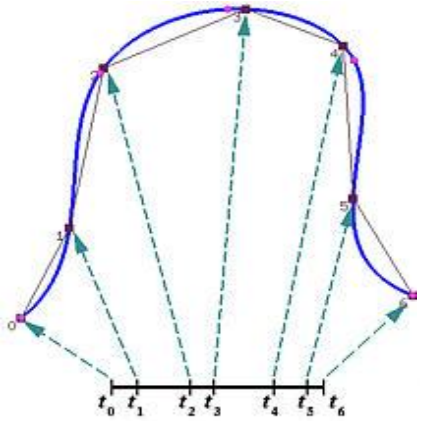
$$C = Z^{-1} D\, N^{-T} \tag{4.28}$$

4.7. Veri Parametrelendirme Yöntemleri

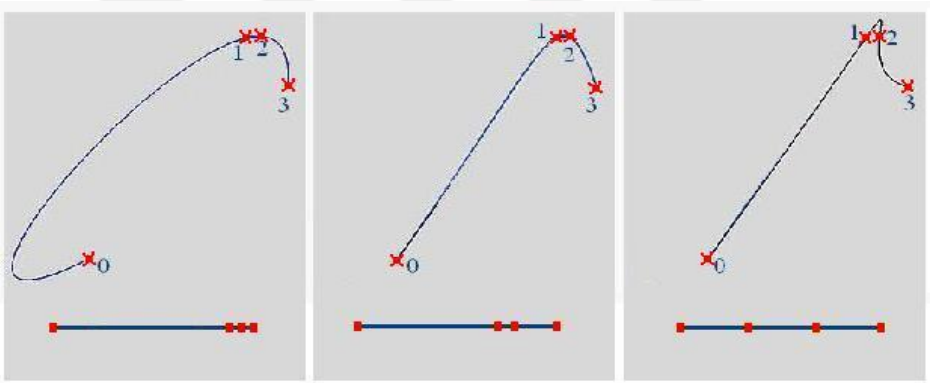
B-spline enterpolasyon ve yaklaşımı problemlerinde, girdi olarak bir nokta kümesi verilir. Düğüm vektörünün [0,1] aralığında değiştiği farz edilir ise, bu aralıktaki bazı kesme -parametre, durak- noktalarının, girdi kümesindeki veri noktalarına karşılık gelmesi istenir. $D_0,...D_n$ veri noktaları için, $t_0,...t_n$ bölgesinde $n+1$ adet parametre değeri tanımlanır. $C(u)$ bütün veri noktalarında geçen bir eğri olarak tanımlı ise, kesim yerlerinde, $0 \le k \le n$ olmak üzere, tamsayı k değerleri için, $D_k = C(t_k)$ olur. Şekil 4.18'de parametrelerin veri noktaları ile ilişkilendirilmesi gösteriliyor.

Parametre değerlerinin seçilmesi belirsizlik içerir ve sonsuz sayıda alternatifler mevcuttur. Bununla birlikte, parametre değerlerinin dikkatsiz bir şekilde seçilmesi, üretilen eğride istenmeyen şekiller ve dalgalanmalar oluşmasına sebep olabilir. Şekil 4.19'da parametre seçimine göre oluşan alternatif eğri enterpolasyonları gösteriliyor. Şekil 4.20'de ise girdi veri noktalarının parametrelendirme sıralamasına göre oluşan çeşitli eğri enterpolasyonları gösteriliyor.

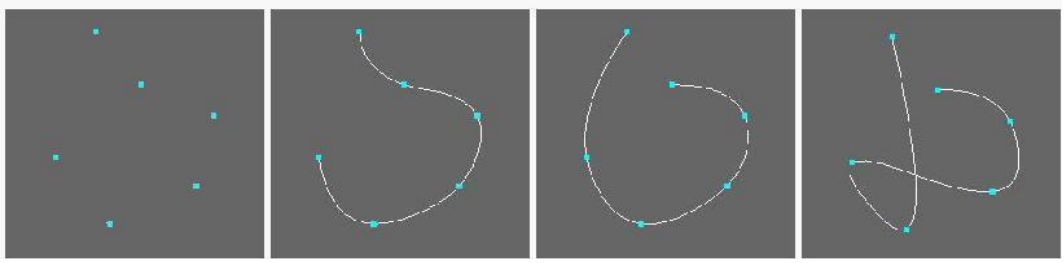
Parametrelendirme aşamasında, uniform, chord length, centripetal, universal olarak isimlendirilen çeşitli metotlar kullanılmaktadır. Sonraki başlıklarda bu metotlar anlatılıyor.



Şekil 4.18. Parametreler ve karşılık gelen veri noktaları [52]



Şekil 4.19. Çeşitli parametrelendirmelere göre üretilen eğriler [52]



Şekil 4.20. Parametrelendirme sırası değişiminin enterpolasyona etkisi [49]

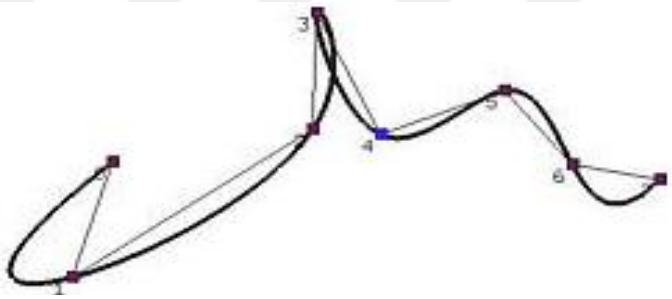
4.7.1. Düzgün dağılımlı parametrelendirme - uniform spaced

Uniform - düzgün dağılımlı parametrelendirme, verilen girdi data noktalarına parametre atamada en basit yöntemdir. Parametrelerin [0,1] aralığında dağıtılacağı

farz edilirse ve $n+1$adet girdi noktanın bu aralığı n parçaya böleceği düşünülür ise, parametreler şu şekilde tanımlanır:

$$t_0 = 0$$
$$t_i = \frac{i}{n},\ 1 \le i \le n-1 \tag{4.29}$$
$$t_n = 1$$

Örneğin, 8 veri noktası verildiğinde, $n=7$ olacağından, uniform parametreler $\left\{0\ \frac{1}{7}\ \frac{2}{7}\ \frac{3}{7}\ \frac{4}{7}\ \frac{5}{7}\ \frac{6}{7}\ 1\right\}$ şeklinde belirlenir. Şekil 4.21'de 8 adet nokta için dügün dağılımlı parametrelendirme ile üretilmiş b-spline eğri gösteriliyor.



Şekil 4.21. Düzgün dağılımlı parametrelendirme [52]

Eğer parametrelendirme [0,1] aralığı yerine [a,b] aralığında yapılmak istenirse, bu aralık eşit parçalara ayrılarak parametre değerleri tespit edilir.

$$t_0 = a$$
$$t_i = a + i\frac{b-a}{n},\ 1 \le i \le n-1 \tag{4.30}$$
$$t_n = b$$

Uniform parametrelendirmenin hesaplaması basit olmasına rağmen eğri enterpolasyonunda istenmeyen sonuçlar üretebilir. Örneğin, girdi veri noktaları, parametre nispetinde düzgün dağılımlı değil ise, istenmeyen çeşitli kıvrımlar oluşabilir. Bu problemi aşmak için çeşitli parametrelendirme metotları geliştirilmiştir. Ancak bu problem sadece uniform parametre için değil, diğer geliştirilmiş parametrelendirme metotları için de söz konusu olabilir.

4.7.2. Kiriş uzunluğuna göre parametrelendirme - chord length

Girdi veri noktaları arasındaki mesafeler düzensiz dağılımlı ise, chord-length metodu, uniform parametrelendirmeye göre daha iyi çalışır.

$D_{0,}\ D_1, \ldots D_n$ data noktaları verilmiş olsun. D_{i-1} noktası ile D_i noktaları arasındaki mesafe $|D_i - D_{i-1}|$ ile ifade edilir. Bu durumda, tüm data noktaları arasındaki kiriş mesafeleri toplamı şu şekilde ifade edilir:

$$L = \sum_{i=1}^{n} |D_i - D_{i-1}| \tag{4.31}$$

Bu durumda, D_0 noktasından D_k noktasına kadar olan kiriş uzunluğu oranı

$$L_k = \frac{\sum_{i=1}^{k} |D_i - D_{i-1}|}{L} \tag{4.32}$$

şeklinde ifade edilir. Normalize edilmiş durumda, [0,1] aralığında parametrelendirme yapıldığı farz edilirse, şu şekilde bir dağılım olur:

$$t_0 = 0$$

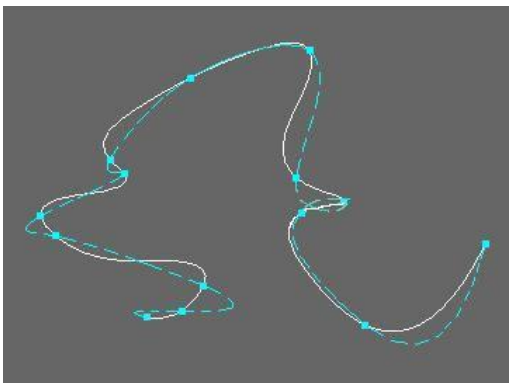
$$t_k = \frac{1}{L} \sum_{i=1}^{k} |D_i - D_{i-1}| \tag{4.33}$$

$$t_n = 1$$

Bu durumda, [0,1] aralığı data noktaları arasındaki mesafe oranında bölgelere paylaştırılmış olur.

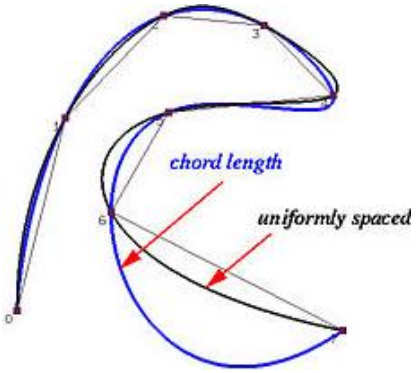
Şekil 4.22'de uniform parametrelendirme (kesik çizgili) ve kiriş uzunluğu parametrelendirmesine göre yapılmış iki enterpolasyon gösteriliyor. Veri noktaları arası mesafeler düzgün dağılımlı olmadığından, düzgün dağılımlı – uniform parametrelendirmenin istenmeyen bazı kıvrımlar oluşturduğu görülüyor. Kiriş uzunluğu – chord parametrelendirme sıklıkla kullanılmaktadır. Ancak polinom

eğrilerin, noktalar arasındaki tarayan eğri uzunluğu ile orantılı şekilde diğer bir ifade ile, birim hıza sahip şekilde parametrelendirmesi mümkün görülmemektedir [53].



Şekil 4.22. Uniform (kesik çizgili) ve chord length parametrelendirme [49]

Noktalar arası kiriş mesafesi, noktalar arasını birleştirecek olan eğri uzunluğuna sadece bir yaklaşım olarak kullanılmaktadır. Bu durumda, daha uzun bir kiriş mesafesi, problemli durumlar üretebilir. Şekil 4.23'de eğrinin son segmentinde, kiriş uzunluğuna göre (chord length) parametrelendirme kullanan eğri, düzgün dağılımlı (uniform) parametrelendir kullanılan eğriye göre daha fazla gereksiz bükülme gösteriyor [52].



Şekil 4.23. Chord length, son segmentte hata [52]

4.7.3. Merkezcil parametrelendirme - centripetal method

Merkezcil parametrelendirme, E. Lee [54] tarafından önerilmiştir. Yarış pistinde araba sürme örneği ile açıklanırsa, pistte araba kullanırken, keskin dönüşlerde merkezkaç kuvvetinin (veya normal kuvvet) çok yüksek olmaması istenir. Güvenli bir sürüş için Lee, yol boyunca merkezkaç kuvvetinin açıdaki değişimle orantılı

olması gerektiğini önermiştir. Centripetal metodu, bu modele bir yaklaşım sunmaktadır. Bu model, chord length metodun geliştirilmiş bir versiyonu olarak görülebilir.

$D_{0,}D_1,\ldots D_n$ data noktaları verilmiş olsun. Üs faktörü $e=1/2$ olarak tanımlanırsa, D_{i-1} noktası ile D_i noktaları arasındaki mesafe $|D_i - D_{i-1}|^e$ ile ifade edilir. Kiriş-chord metodunda bu ifade üssüz olarak kullanılıyordu. Bu durumda, tüm data poligonunun uzunluğu merkezcil - centripetal metoda göre şu şekilde ifade edilir:

$$L = \sum_{i=1}^{n} |D_i - D_{i-1}|^e \tag{4.34}$$

D_0 noktasından D_k noktasına kadar olan data poligon uzunluğunun toplam poligon uzunluğuna oranı ise

$$L_k = \frac{\sum_{i=1}^{k} |D_i - D_{i-1}|^e}{L} \tag{4.35}$$

şeklinde ifade edilir. Normalize edilmiş durumda, [0,1] aralığında parametrelendirme yapıldığı farz edilirse, t_k değerleri şu şekilde bir dağılıma uğrar:

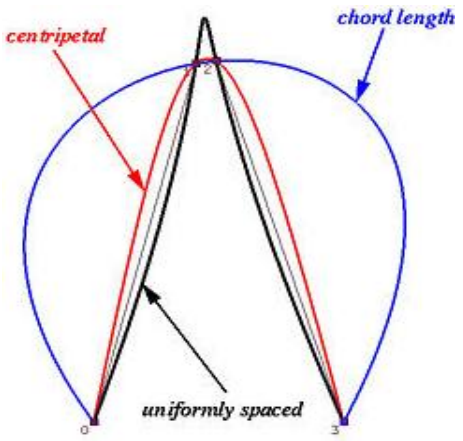
$$t_0 = 0$$

$$t_k = \frac{1}{L} \sum_{i=1}^{k} |D_i - D_{i-1}|^e \tag{4.36}$$

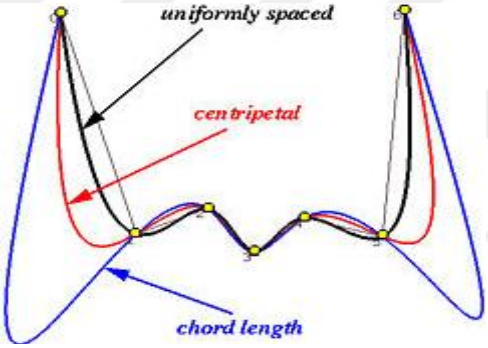
$$t_n = 1$$

Eğer $e=1$ seçilirse merkezcil-centripetal parametrelendirme, kiriş-chord parametrelendirmeye indirgenmiş olur. Eğer $e<1$, örneğin $e=1/2$ (karekök) seçilir ise, kiriş_uzunluğu>1 farz edilmek üzere, $|D_k - D_{k-1}|^e$ değeri, $|D_k - D_{k-1}|$ değerinden küçük hale geleceğinden, data poligonundaki uzun kirişlerin etkisi azaltılmış olur ve kısa kirişlerin etkisi arttırılmış olur. Bu davranıştan dolayı Lee merkezcil-centripetal metodun keskin dönüşlerde kiriş-chord metoduna göre daha iyi davranış gösterdiğini

söylemektedir. Şekil 4.24 ve 4.25'de bu metotların çeşitli veri durumlarında karşılaştırılması gösteriliyor.



Şekil 4.24. Parametrelendirme metotları [52]



Şekil 4.25. Parametrelendirme metotları [52]

4.7.4. Düğüm vektörü üretimi

Parametre değerlerinin üretilmesinden sonra, bu değerlerden yola çıkılarak düğüm vektörü üretilir. Elimizde $t_0, t_1, ...t_n$ şeklinde tanımlı $n+1$ adet parametre değeri bulunduğunu ve p 'inci derece b-spline parçaları kullanılacağını farz edersek, $m = n + p + 1$ şeklinde tanımlı olmak üzere, $m+1$ adet düğüm değerine ihtiyacımız olur. Eğer eğri clamped – sıkılmış olarak tanımlı ise, ilk baştaki $p+1$ adet düğüm değeri 0 ve sondaki $p+1$ adet düğüm değeri 1 olur. Ortadaki $n-p$ adet düğüm ise, düzgün dağılımlı – uniform olarak veya çeşitli metotlara göre dağıtılır.

Uniform-düzgün dağılımlı parametrelendirme kullanılır ise, ortanca terimler $n-p+1$ adet parçaya bölünür.

$$u_0 = u_1 = ...u_p = 0$$

$$u_{j+p} = \frac{j}{n-p+1}, \quad j = 1,2,...,n-p \tag{4.37}$$

$$u_{m-p} = u_{m-p+1} = ...u_m = 1$$

Örneğin, 6 adet parametre için ($n=5$) ve 3. derece b-spline için ($p=3$), $(n+p+1)+1=(5+3+1)+1=10$ adet düğüm değeri ($m=9$) gereklidir. Clamped-sıkılmış b-spline durumunda, baştaki ve sondaki $p+1$ adet düğüm değerleri 0 ve 1 olacağından, düğüm vektörü şu şekilde oluşur: $\{0\ 0\ 0\ 0\ u_4\ u_5\ 1\ 1\ 1\ 1\}$. Ortanca iki terim, [0,1] aralığını üç eşit parçaya böler. Bu durumda düğüm vektörü $\left\{0\ 0\ 0\ 0\ \frac{1}{3}\ \frac{2}{3}\ 1\ 1\ 1\ 1\right\}$ şeklinde oluşur.

Uniform – düzgün dağılımlı düğüm vektörünün belirlenmesi için, parametrelendirme değerlerinin bilinmesi gerekmez. Hesaplanması basit olmasına rağmen, bu şekilde üretilen düğüm vektörü, chord – kiriş mesafesine göre parametrelendirme ile birlikte kullanıldığında enterpolasyon matrisleri singular – tekil duruma geleceğinden çözümsüz bir durum oluşur. Bu durumdan sakınmak için, Piegl [48] tarafından önerilen ortalama metodu kullanılarak, parametre değerlerinin kayan ağırlıklı ortalamaları, düğüm vektörünü teşkil eder.

Parametrelerin ortalamasına göre düğüm vektörü üretme işlemi Denklem (4.38) deki gibi ifade edilebilir. Örneğin, üçüncü derece b-spline için, $p=3$, on adet düğüm değeri üretilecek (m=9) olsun. Parametre değerleri, $\{t_0\ t_1\ t_2\ t_3\ t_4\ t_5\} = \left\{0\ \frac{1}{4}\ \frac{1}{3}\ \frac{2}{3}\ \frac{3}{4}\ 1\right\}$ olarak verilmiş olsun. Bu durumda,

$$u_0 = u_1 = ...u_p = 0$$

$$u_{j+p} = \frac{1}{p}\sum_{i=j}^{j+p-1} t_i\ , \quad j = 1,2,...,n-p \tag{4.38}$$

$$u_{m-p} = u_{m-p+1} = ...u_m = 1$$

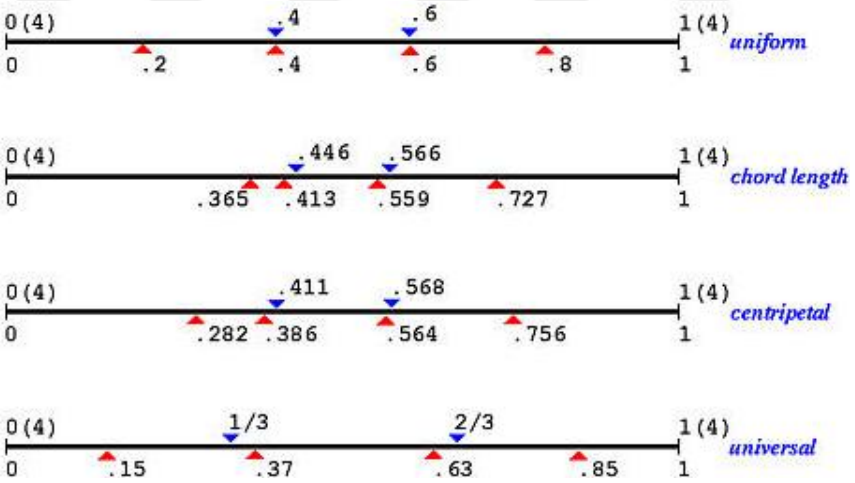
$$u_0 = u_1 = u_2 = u_3 = 0,\quad u_4 = \frac{\frac{1}{4}+\frac{1}{3}+\frac{2}{3}}{3} = \frac{5}{12},\quad u_5 = \frac{\frac{1}{3}+\frac{2}{3}+\frac{3}{4}}{3} = \frac{7}{12}\quad u_6 = u_7 = u_8 = u_9 = 1$$

değerlerini alır. Şekil 4.26'da verilen parametre değerlerine karşılık ortalama metoduna göre üretilen düğüm değerleri gösteriliyor.



Şekil 4.26. Parametre ve düğüm değerleri [52]

Clamping-sıkma uygulanan 3. derece b-spline için, uniform-düzgün dağılımlı, chord-kiriş mesafesi ve centripetal-merkezcil parametrelendirmeler için, ortalamaya göre üretilen düğüm vektörleri şekil 4.27'de gösteriliyor.



Şekil 4.27. Seçilen parametreler (alt) ve ortalama ile düğüm vektörleri (üst) [52]

4.7.5. Evrensel metot - universal method

Choong Gyoo Lim, 1999 yılında diğer metotlardan farklı işleyen bir metot önerdi [55]. Önceki metotlarda, parametre değerleri belirlendikten sonra düğüm vektörü bu parametre değerlerinden üretilirken, yeni önerilen bu metot, uniform – düzgün dağılımlı clamped-sıkılmış düğüm vektöründen yola çıkarak parametre değerlerini hesaplama üzerine kurgulanmıştır.

Lim'in [55] metodunda, düzgün dağılımlı bir düğüm vektöründen yola çıkılarak, baz fonksiyonlar üretilir. Bu baz fonksiyonların tepe değerleri, parametre listesine girdi olarak kabul edilir. Bu metot, önceki metotlara göre daha doğal görünümlü eğriler üretmesine rağmen, bazı durumlarda istenmeyen salınımlar üretmektedir.

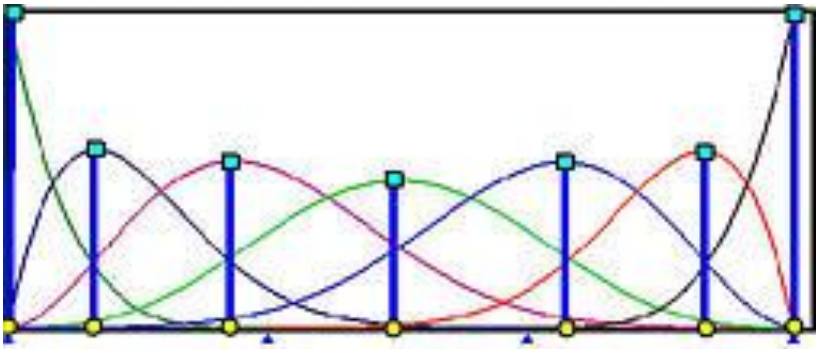
p 'inci dereceden b-spline kullanılarak, $n+1$ adet data noktası için gerekli parametre değerleri üretilmek istensin. $m=n+p+1$ olmak üzere, düğüm vektöründeki eleman sayısı $m+1$ olur. Bu durumda düzgün dağılımlı, sıkılmış-clamped düğüm vektörü şu şekilde elde edilir:

$$u_0 = u_1 = ...u_p = 0$$

$$u_{p+i} = \frac{i}{n-p+1}, \quad i = 1,2,...,n-p \tag{4.39}$$

$$u_{m-p} = u_{m-p+1} = ...u_m = 1$$

Bu düğüm vektörü ile elde edilen baz fonksiyonların maksimum yaptığı yerler, parametre değerleri olarak seçilir. Bu durum şekil 4.28'de gösteriliyor.



Şekil 4.28. Baz fonksiyon tepeleri – parametreler [52]

4.7.6. Foley-Nielson metodu

Foley ve Neilson [56] tarafından önerilen metot. Önceki metotlar, data noktaları arasındaki kiriş mesafesini kullanırken, Foley-Nielson metodu, komşu noktalar arasındaki açıyı da hesaba katmaktadır. Bu metot, noktalar arasındaki mesafeyi ölçmek üzere Nielson (1987) ölçütünü kullanır. Bu sayede, döndürme, öteleme ve skalalandırma işlemlerine girdiğinde, geometride değişiklik üretmez. Dönüştürülmüş b-spline eğri, dönüştürülmüş data noktalarından elde edilebilir.

$m = 2$, iki boyutlu data noktaları ile çalışılması durumunda, $Q = \{q_{ij}\}, i, j = 1,2$ katsayı matrisi, şu şekilde tanımlanır:

$$q_{11} = \frac{V_y}{g},\ q_{22} = \frac{V_x}{g},\ q_{12} = q_{21} = -\frac{V_{xy}}{g}$$

$$g = V_x V_y - (V_{xy})^2,$$

$$V_x = \frac{\sum_{i=0}^{n}(x_i - \bar{x})^2}{n+1},\ V_y = \frac{\sum_{i=0}^{n}(y_i - \bar{y})^2}{n+1}, \tag{4.40}$$

$$V_{xy} = \frac{\sum_{i=0}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n+1}$$

$$\bar{x} = \frac{\sum_{i=0}^{n} x_i}{n+1},\ \bar{y} = \frac{\sum_{i=0}^{n} y_i}{n+1}$$

Bu durumda U ve V noktaları arasındaki Neilson mesafesi şu şekilde tanımlanır:

$$M[P](U,V) = \sqrt{(U-V)Q(U-V)^T} \tag{4.41}$$

Hesaplanan Neilson mesafesine göre, chord-kiriş mesafesi parametrelendirmesi yapılır ise, affine-invariant, transformasyonlara girebilen bir parametrelendirme elde edilebilir.

Foley parametrelendirmesinde, parametre adımı Δt_i şu şekilde tanımlanır [55, 56, 50]:

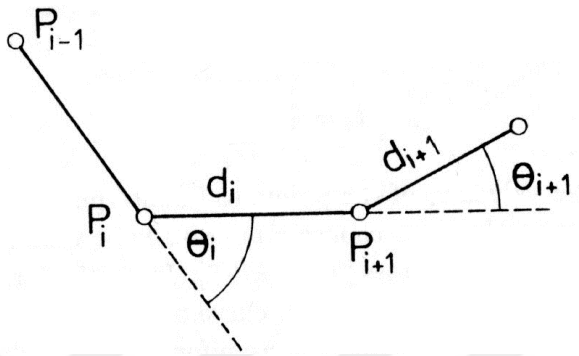
$$\Delta t_0 = d_0\left[1 + \frac{3\hat{\theta}_1 d_1}{2(d_0 + d_1)}\right]$$

$$\Delta t_i = d_i\left[1 + \frac{3\hat{\theta}_i d_{i-1}}{2(d_{i-1} + d_i)} + \frac{3\hat{\theta}_{i+1} d_{i+1}}{2(d_i + d_{i+1})}\right],\ i = 1,2....,n-2 \tag{4.42}$$

$$\Delta t_{n-1} = d_{n-1}\left[1 + \frac{3\hat{\theta}_{n-1} d_{n-2}}{2(d_{n-2} + d_{n-1})}\right]$$

$d_i = M[P](P_i, P_{i+1})$, $\hat{\theta}_i = \min(\theta_i, \frac{\pi}{2})$ ve $\theta_i = \pi - \arccos\left[\frac{d_{i-1}^2 + d_i^2 + M^2[P](P_{i-1}, P_{i+1})}{2d_i d_{i-1}}\right]$.

Burada, θ_i de Neilson mesafesi ile hesaplandığından, transformasyonlara girdiğinde etkilenmeyen, affine invariant bir açıdır. Şekil 4.29'da Foley metodunda kullanılan notasyonlar gösteriliyor.



Şekil 4.29. Foley metodunda kullanılan parametreler [50]

4.7.7.Önerilen metot: dinamik merkezcil metot (dynamic centripetal parameterization)

Lee tarafından önerilen merkezcil parametrelendirme metodu, CAD-CAM uygulamalarında, enterpolasyon amaçlı olarak sıklıkla kullanılmaktadır [54]. Bu metot halihazırda, her bir kiriş mesafesi için aynı üs değerini kullanmaktadır. Yaptığımız deneylerde, üs değerinin, kiriş mesafesine göre dinamik olarak değiştirilmesinin, parametrelendirme başarımını arttırdığı görülmüş ve kiriş mesafesine göre üs değerini dinamik olarak değiştiren bir hesap metodu geliştirilmiştir.

Dinamik üs uyarlaması katılarak, merkezcil metodun ifadeleri tekrar ele alınır ise, $D_0, D_1, \ldots D_n$ data noktaları verilmiş olsun. Üs faktörü, her bir kiriş için dinamik olarak değişmek üzere, e_i olarak tanımlanırsa, D_{i-1} noktası ile D_i noktaları arasındaki, parametre dağılımına baz olan mesafe $|D_i - D_{i-1}|^{e_i}$ ile ifade edilir. Bu durumda, tüm data poligonunun uzunluğu önerilen metoda göre şu şekilde ifade edilir:

$$L = \sum_{i=1}^{n} \left| D_i - D_{i-1} \right|^{e_i} \tag{4.40}$$

D_0 noktasından D_k noktasına kadar olan data poligon uzunluğunun toplam poligon uzunluğuna oranı ise

$$L_k = \frac{\sum_{i=1}^{k} \left| D_i - D_{i-1} \right|^{e_i}}{L} \tag{4.41}$$

şeklinde ifade edilir. Normalize edilmiş durumda, [0,1] aralığında parametrelendirme yapıldığı farz edilirse, t_k değerleri şu şekilde bir dağılıma uğrar:

$$\begin{aligned} t_0 &= 0 \\ t_k &= \frac{1}{L} \sum_{i=1}^{k} \left| D_i - D_{i-1} \right|^{e_i} \\ t_n &= 1 \end{aligned} \tag{4.42}$$

Merkezcil – centripetal metotda, üs değeri sabit bir değer alınmakta, genellikle $e = 1/2$ olarak seçilmektedir. Yaptığımız enterpolasyon deneylerinde, centripetal metodun, uzun kirişler ve kısa kirişler için aynı üs değerini kullanarak, uniform ve chord modele göre nispeten adaletli olsa bile, üs değerinin sabitliğinden kaynaklanan adaletsiz bir parametre dağılımı yaptığını gözlemledik. Bunun çözümü olarak, $e < 1$ olmak üzere ve tüm kiriş mesafelerinin uzunluğu 1'den büyük olmak üzere, kiriş mesafesinin logaritması ile ters orantılı olarak, üs değerini belirli aralıkta dağıtan bir metot geliştirildi.

Her bir adımlamada, i'inci kiriş için kullanılacak olan, kiriş mesafesine göre değişen dinamik üs değeri e_i'nin hesaplanması, bir sonraki aşamada anlatılıyor.

$chord_i$: i'inci kiriş mesafesi $\left| D_i - D_{i-1} \right|$ olmak üzere,

$chord_{\max}$: en uzun kiriş mesafesi, $chord_{\min}$: en kısa kiriş mesafesi olarak tanımlanırsa ve izin verilen en büyük üs değeri $e_{\max}$, izin verilen en küçük üs değeri

ise $e_{\min}$ olarak tanımlanır ise, i'inci kirişe uygulanacak olan e_i dinamik üs değeri Denklem (4.43) ve Denklem (4.44) ile elde edilir.

$$\frac{(e_{\max} - e_{\min})}{\log(chord_{\max}) - \log(chord_{\min})} = \frac{(e_i - e_{\min})}{\log(chord_{\max}) - \log(chord_i)} \tag{4.43}$$

$$e_i = \frac{(\log(chord_{\max}) - \log(chord_i))\ (e_{\max} - e_{\min})}{\log(chord_{\max}) - \log(chord_{\min})} + e_{\min} \tag{4.44}$$

Her bir kiriş için, özel üretilmiş üs değeri e_{i} kullanılarak, uzun mesafeli kirişlerde, parametre ilerlemesi dengeli bir şekilde kısaltılarak b-spline eğri üzerinde germe etkisi oluşturulur. Kısa mesafeli kirişlerde ise, toplam dengeyi bozmayan bir miktar yayma etkisi üretilir. En uzun mesafeli kirişte $e_{min} = 0{,}35$ üs değeri kullanılır iken, en kısa mesafeli kiriş için, $e_{max} = 0{,}65$ üs değeri kullanılır. Diğer kirişler için, üs değerleri e_{max} ve e_{min} değerleri arasında, ilgili kiriş mesafesinin logaritması ile ters orantılı olarak dağıtılır. Kiriş mesafeleri 1'den büyük sayılar olması şartından dolayı, uzun mesafeli kirişlerin parametresi kısaltılmış olur.

Bir sonraki adımda, deneyler neticesi, çeşitli parametrelendirme metotları ve önerilen metot ile üretilen eğriler kıyaslanmakta ve eğri enterpolasyon hata miktarındaki iyileştirmeler, grafikler ile gösterilmektedir.

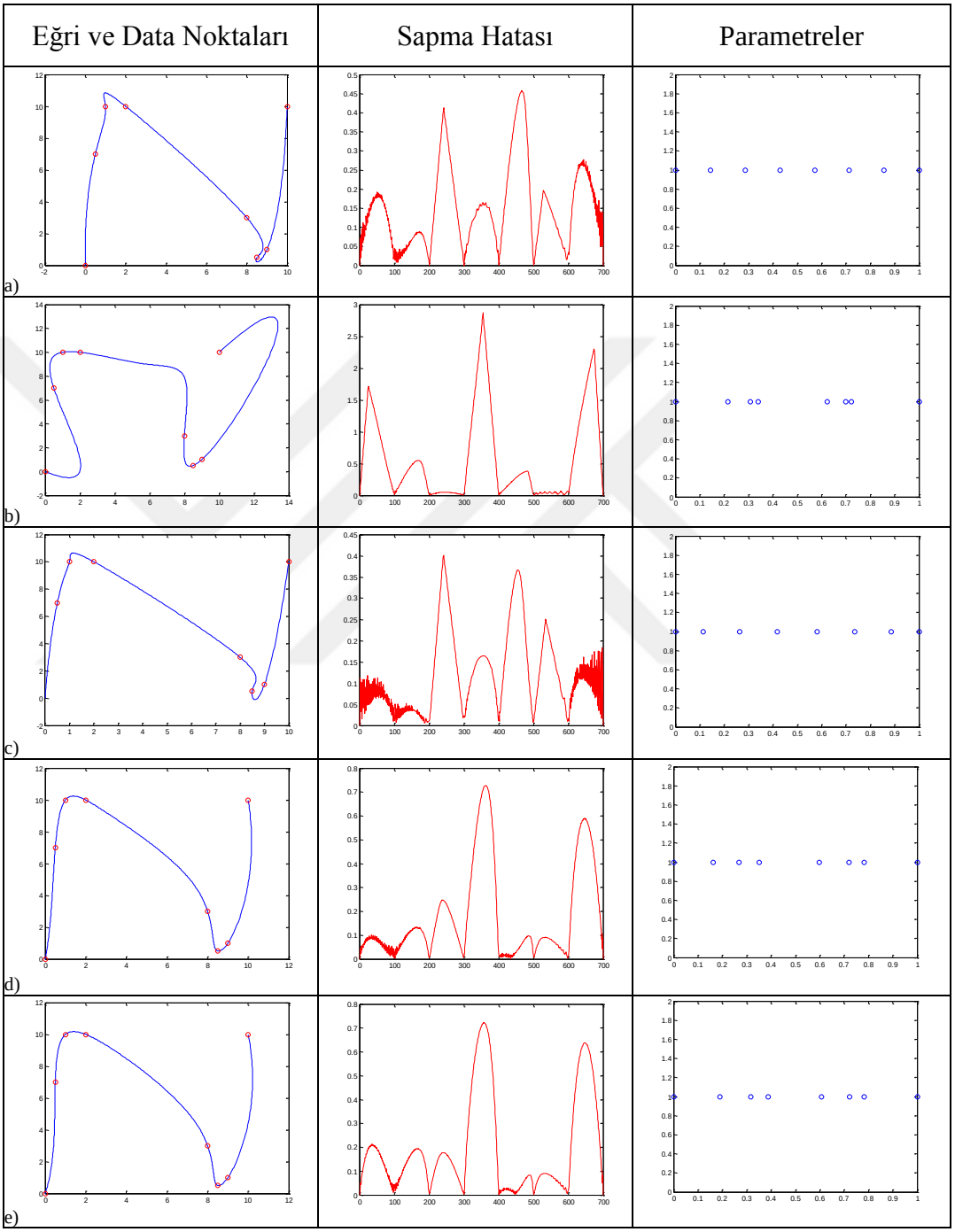
4.7.8.Uniform – chord – centripetal – universal - önerilen metot: eğri kıyaslamaları

Hata ölçütü olarak, Fang ve Hung'un [57] çalışmasında bahsedilen, eğrinin her bir adımda, data noktalarını birleştiren kirişten uzaklığını ölçme tekniği kullanılmıştır. Denklem (4.45)'de hata ölçütünün matematiksel ifadesi gösteriliyor.

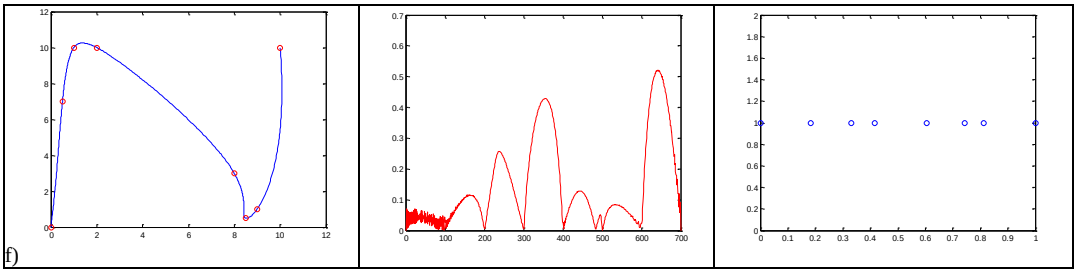
$$D_i = \max_{\bar{u}_i < u < \bar{u}_{i+1}} dist(C(t), L_i),\ 0 \le i \le n-1 \tag{4.45}$$

Burada $dist(C(t), L_i)$ ifadesi parametrenin t anındaki değerinin, L kirişine olan mesafesini gösteriyor.

Şekil 4.30 - 4.34 arasında, farklı data setler için, farklı metotlar ile üretilen kübik b-spline eğriler ve hata fonksiyonları, parametreler ile birlikte karşılaştırmalı olarak gösteriliyor. Sapma hataları ise Tablo 4.1 - 4.4 içinde gösterilmiştir.



Şekil 4.30. Birinci data set ile kübik eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot

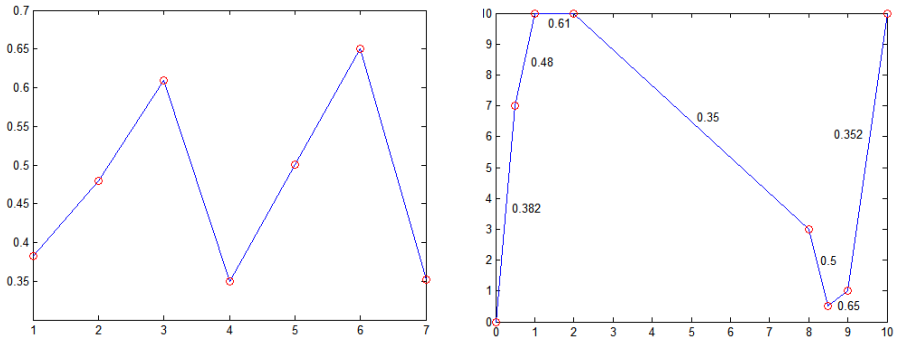

f)

Şekil 4.30. (Devam) Birinci data set ile kübik eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot

Şekil 4.30'da enterpole edilen birinci data set, Larry Irvine, Samuel Marin ve Philip Smith'in çalışmasından alınmıştır [58]. Bu veriler, ani yön değişimlerinden ve seyrek data noktalarından dolayı enterpolasyon metotları için problem üretebilir zorluktadır.

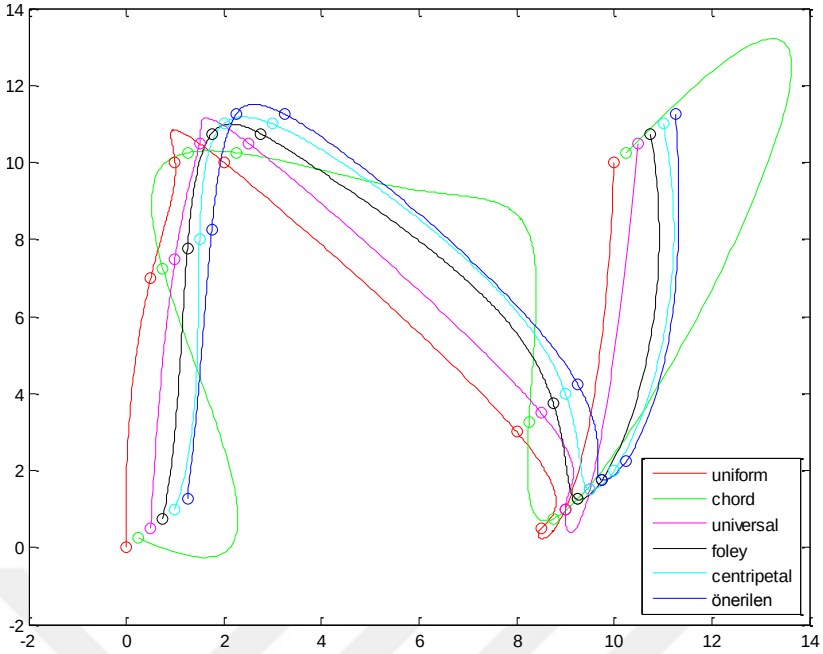
Şekil 4.31'de birinci data set için kullanılan e_i üs değerleri, kiriş sırasına göre ve kirişler üzerinde gösterilmiştir. Uzun mesafeli kirişlerde üs değeri düşürülerek, eğri üzerinde germe etkisi üretilmiştir.

Şekil 4.32'de çeşitli metotlara ve önerilen metoda göre, birinci data set ile üretilen eğriler aynı grafik üzerinde gösteriliyor. Grafikte, uniform, chord length ve universal metotlar tarafından üretilen eğrilerin, istenmeyen salınımlar ürettiği görülmektedir.



Şekil 4.31. Birinci data set için e_i üs değerleri

Tablo 4.1'de birinci data set için üretilen eğrilerin, kirişten maksimum sapmaları, hata ölçütü olarak gösterilmiştir. Maksimumların ortalamasına bakıldığında, önerilen metot başarılı eğriler arasında en düşük hatayı üretmektedir. Universal metodun ortalama hatası düşük görünmesine rağmen, başarılı bir eğri üretememiştir. Centripetal metodun ortalama hatası ve RMS hata değeri, önerilen metodun dinamik üs geliştirmesi ile Foley metodundan başarılı hale gelmiştir.



Şekil 4.32. Birinci data set için metotlara göre eğriler

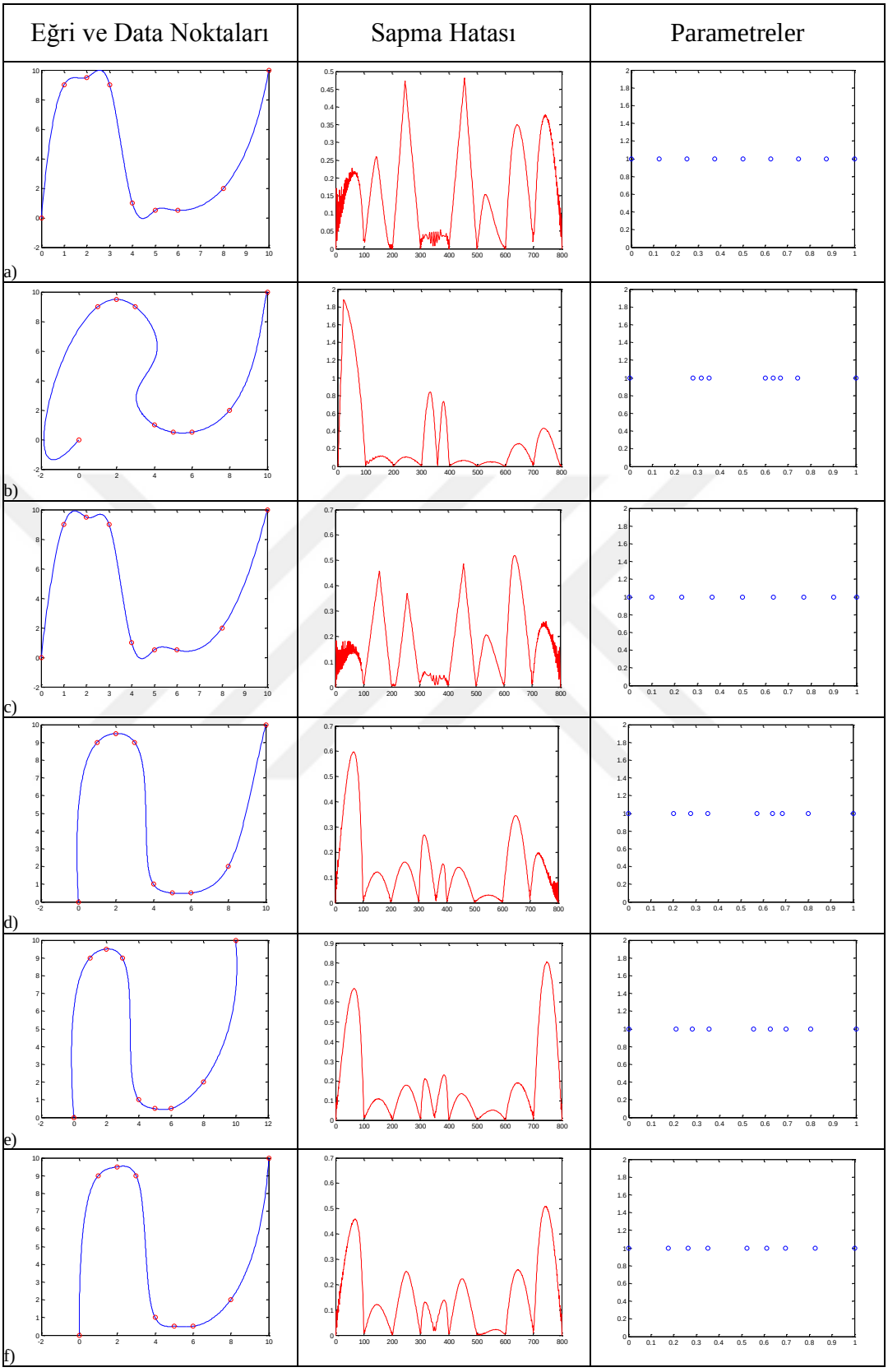
Tablo 4.1. Şekil 4.30'daki eğri segmenti boyunca kirişten sapmalar

Metot	1	2	3	4	5	6	7	Ortalama	RMS	Maks. Sapma	Süre (ms)
Uniform	0,1918	0,0889	0,4133	0,1647	0,4564	0,1964	0,2765	0,255	0,2843	0,4564	49,860
Chord	1,7113	0,5507	0,0524	2,877	0,3785	0,061	2,304	1,133	1,5569	2,877	50,739
Universal	0,102	0,047	0,4007	0,1642	0,3667	0,2513	0,1360	0,2097	0,2439	**0,4007**	135,781
Foley	0,0988	0,1354	0,2475	0,728	0,0976	0,0909	0,5904	0,2840	0,3753	0,728	77,506
Centripetal	0,2135	0,1952	0,1777	0,7214	0,0832	0,0906	0,6385	0,3028	0,3889	0,7214	**49,789**
Önerilen	0,0369	0,1147	0,2573	0,4283	0,1287	0,084	0,521	**0,2244**	0,2826	**0,521**	52,765

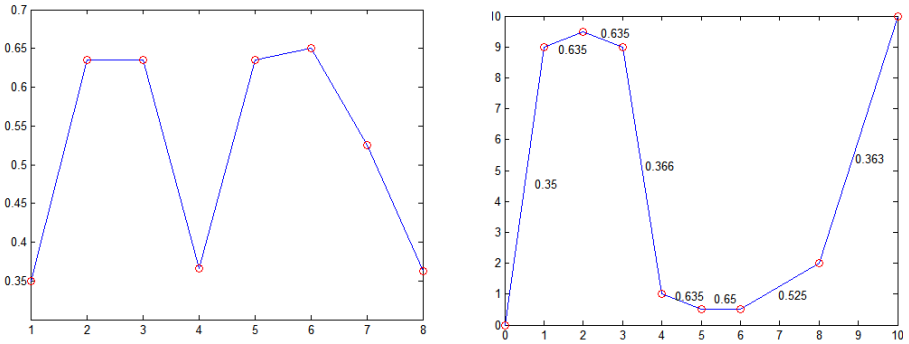
Şekil 4.33'de ikinci data set için çeşitli metotlar parametrelendirme metotları ile üretilen kübik b-spline eğriler ve eğri segmentlerinin kirişten sapma fonksiyonları, seçilen parametreler ile birlikte gösteriliyor.

Şekil 4.33'deki ikinci data set, birinci data set ile aynı çalışmadan alınmıştır [58]. Benzer şekilde ani yön değişimleri ve seyrek data zorluğu burada da mevcuttur.

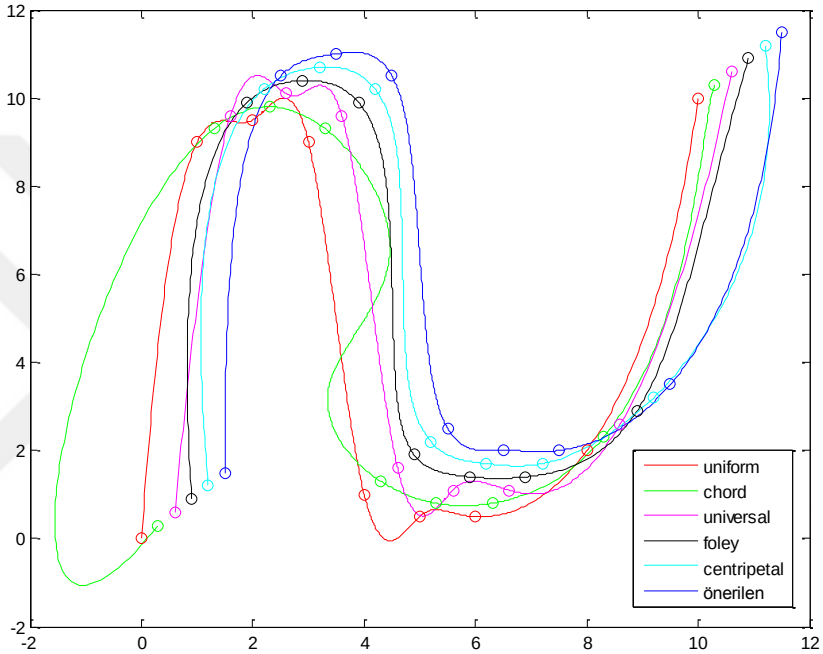
Şekil 4.34'de ikinci data set için kullanılan e_i üs değerleri, kiriş sırasına göre ve kirişler üzerinde gösterilmiştir. Şekil 4.35'de çeşitli metotlara ve önerilen metoda göre, ikinci data set ile üretilen eğriler grafik üzerinde gösteriliyor. Centripetal ve önerilen metotların eğri enterpolasyon başarımının diğer metotlardan yüksek olduğu görülmektedir.



Şekil 4.33. İkinci data set ile kübik eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot



Şekil 4.34. İkinci data set için e_i üs değerleri



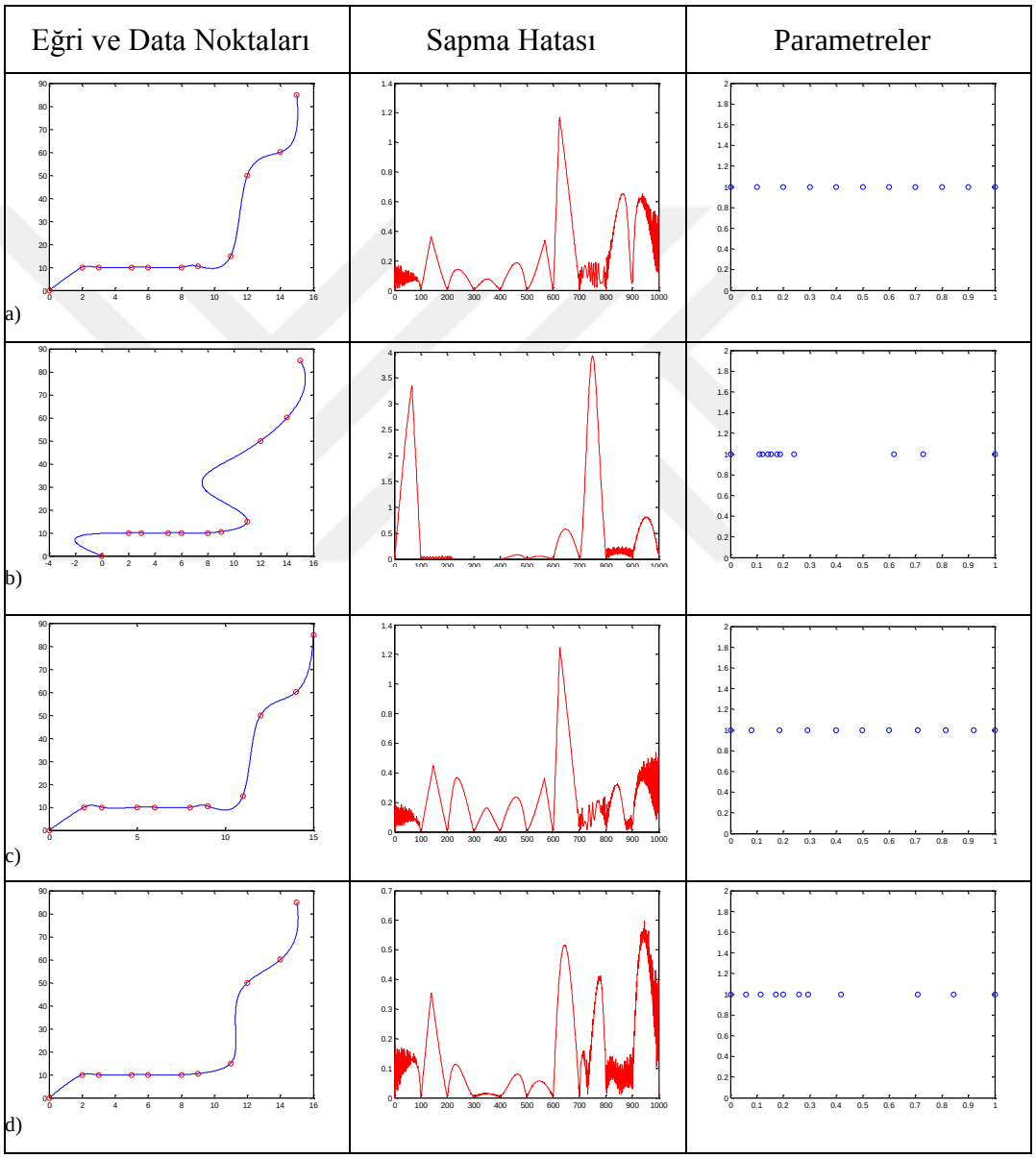
Şekil 4.35. İkinci data set için metotlara göre eğriler

Tablo 4.2. Şekil 4.33'deki eğri segmenti boyunca kirişten sapmalar

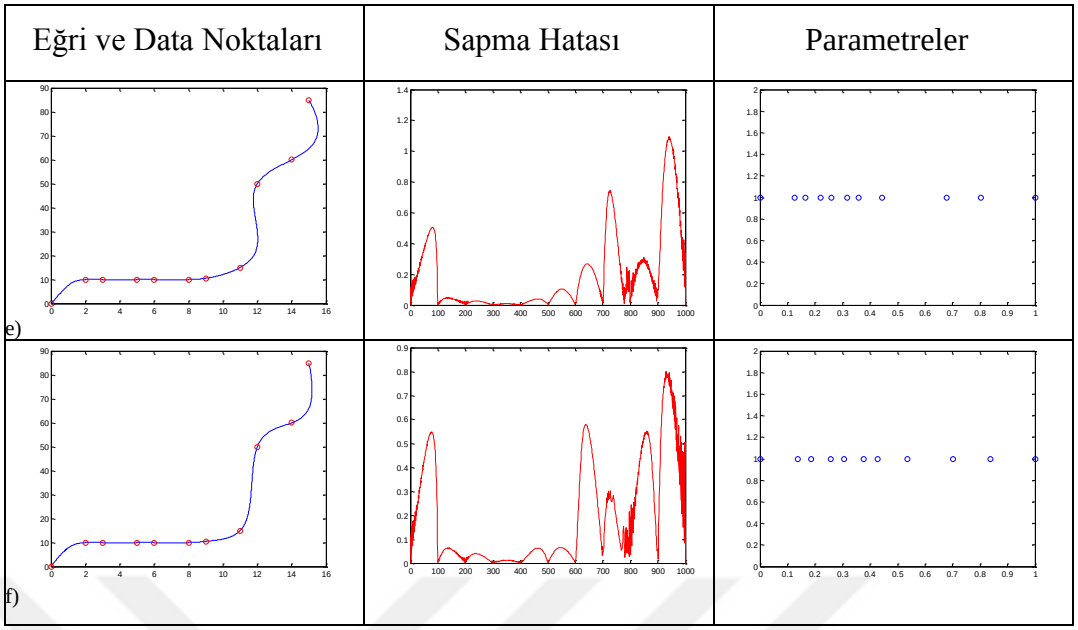
Metot	1	2	3	4	5	6	7	8	Ortalama	RMS	Maks. Sapma	Süre (ms)
Uniform	0,209	0,259	0,473	0,049	0,482	0,152	0,35	0,376	0,2938	0,3271	**0,4733**	50,461
Chord	1,847	0,118	0,106	0,841	0,067	0,053	0,259	0,431	0,4652	0,7420	1,847	**49,805**
Universal	0,18	0,442	0,367	0,054	0,476	0,193	0,519	0,256	0,3109	0,3471	0,5187	133,821
Foley	0,59	0,12	0,161	0,269	0,141	0,031	0,345	0,199	**0,2319**	0,2829	0,5895	76,887
Centripetal	0,667	0,109	0,177	0,231	0,134	0,051	0,191	0,802	0,2952	0,3940	0,8024	52,574
Önerilen	0,46	0,121	0,251	0,14	0,223	0,023	0,258	0,509	**0,248**	0,2926	**0,5094**	55,919

Tablo 4.2'de ikinci data set için üretilen eğrilerin, kirişten maksimum sapmaları, hata ölçütü olarak gösterilmiştir. Ortalama hatalara bakıldığında, önerilen metot Foley metodunun başarımına yakın bir hata üretmektedir. Uzun kirişlerde oluşan maksimum sapma ise önerilen metot ile düşük hale gelmektedir.

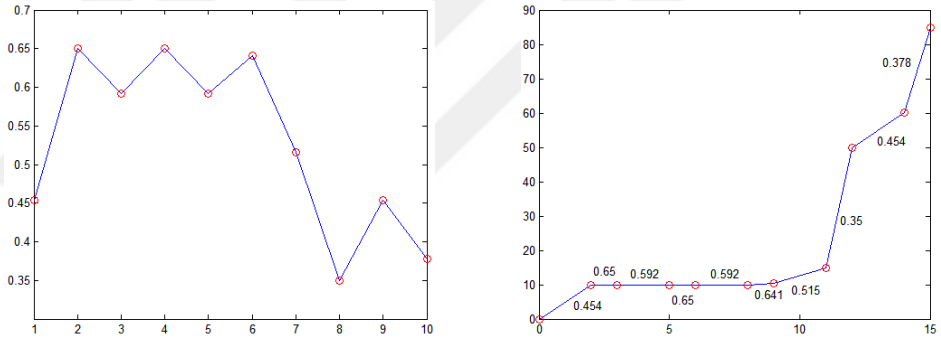
Şekil 4.36'daki üçüncü data set, Fred Fritsch ve Ralph Carlson'un çalışmasından alınmıştır [59]. Burada ani mesafe değişiklikleri test edilmektedir. Şekil 4.37'de üçüncü data set için kullanılan e_i üs değerleri, kiriş sırasına göre ve kirişler üzerinde gösterilmiştir. Şekil 4.38'de çeşitli metotlara ve önerilen metoda göre, üçüncü data set ile üretilen eğriler gösteriliyor. Bu data set için Foley ve önerilen metotların eğri enterpolasyon başarımının diğer metotlardan yüksek olduğu görülmektedir.



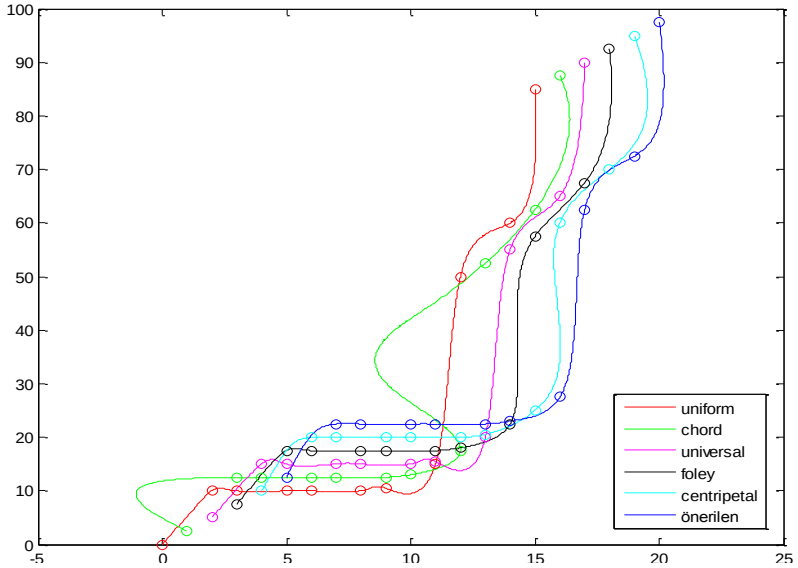
Şekil 4.36. Üçüncü data set ile kübik eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot



Şekil 4.36. (Devam) Üçüncü data set ile kübik eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot



Şekil 4.37. Üçüncü data set için e_i üs değerleri



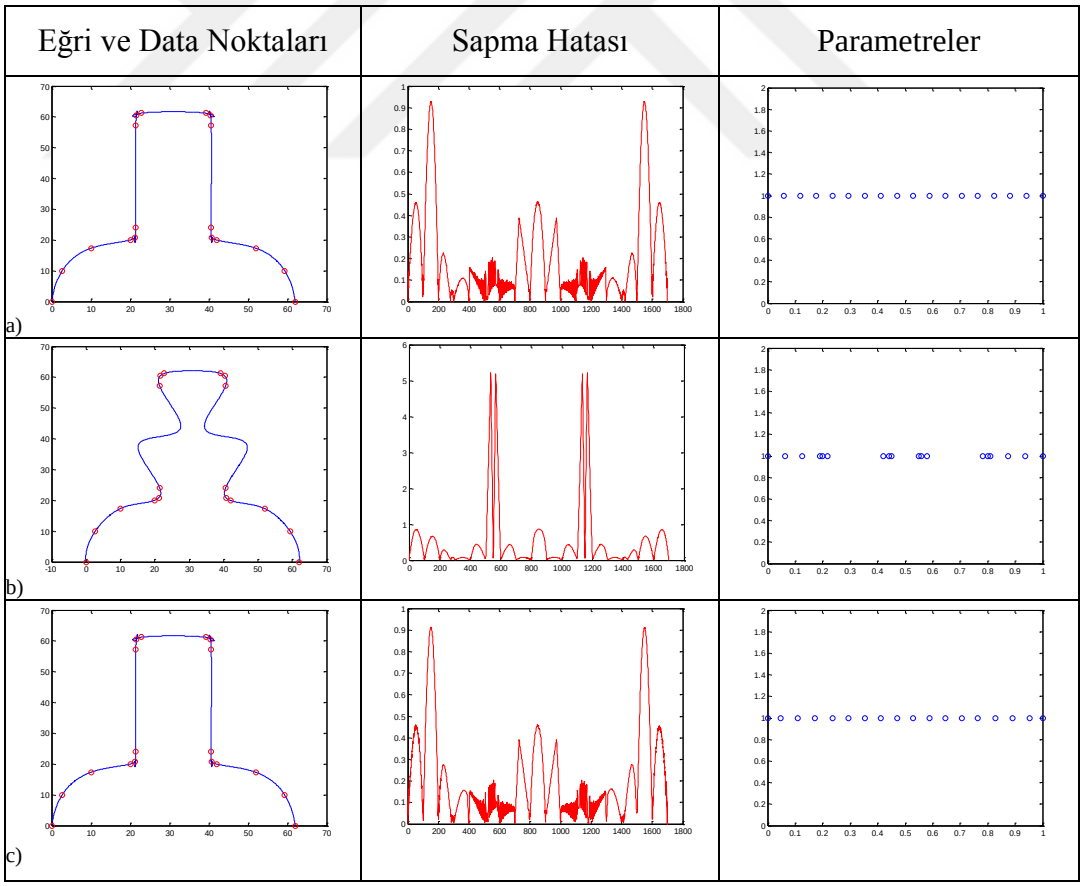
Şekil 4.38. Üçüncü data set için metotlara göre eğriler

Tablo 4.3. Şekil 4.36'daki eğri segmenti boyunca kirişten sapmalar

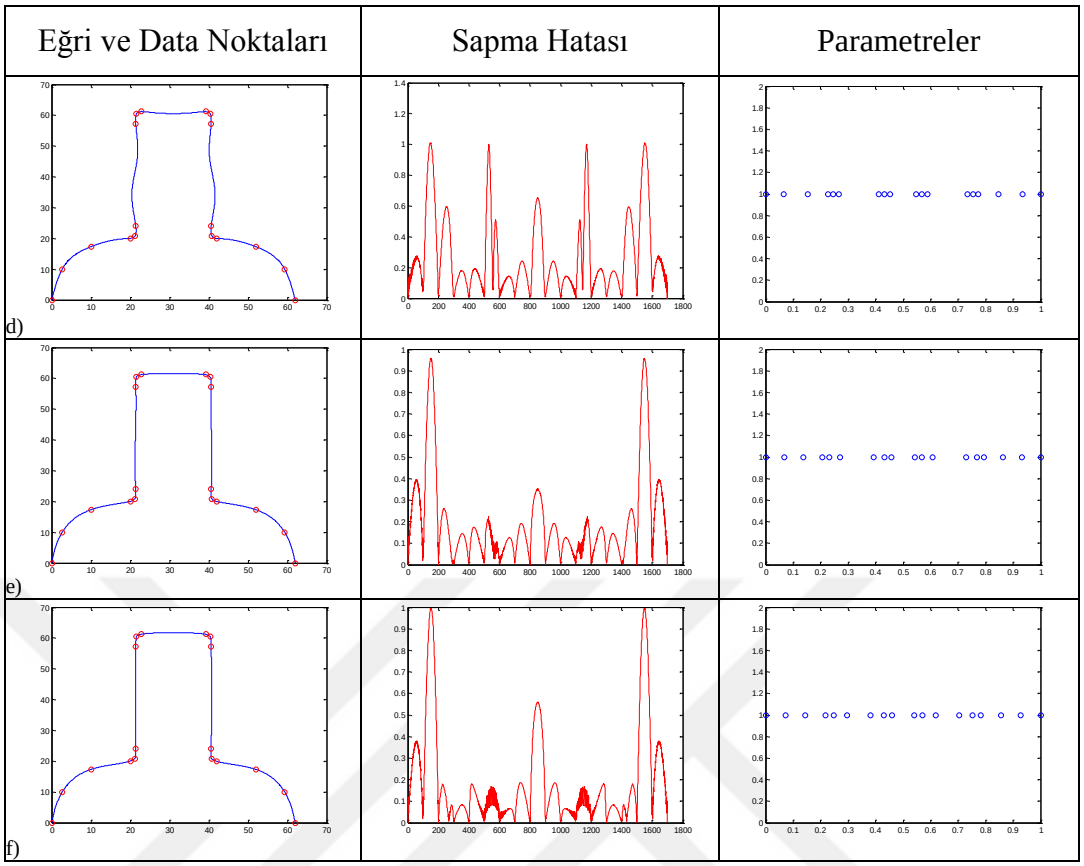
Metot	1	2	3	4	5	6	7	8	9	10	Ort.	RMS	Maks. Sapma	Süre (ms)
Uniform	0,12	0,359	0,142	0,078	0,185	0,337	1,168	0,19	0,652	0,65	0,38	0,50	1,16	**46**
Chord	3,34	0,057	0,011	0,008	0,082	0,053	0,583	3,915	0,238	0,80	0,91	1,66	3,91	49
Universal	0,16	0,447	0,366	0,162	0,234	0,361	1,246	0,212	0,324	0,46	0,39	0,49	1,24	135
Foley	0,16	0,354	0,111	0,131	0,079	0,057	0,516	0,412	0,13	0,59	0,25	0,31	**0,59**	75
Centripetal	0,50	0,045	0,027	0,001	0,040	0,105	0,268	0,741	0,305	1,09	0,31	0,46	1,09	49
Önerilen	0,57	0,063	0,04	0,013	0,063	0,065	0,576	0,282	0,551	0,79	**0,30**	0,41	0,79	55

Tablo 4.3'de üçüncü data set için üretilen eğrilerin, kirişten maksimum sapmaları gösterilmiştir. Önerilen metot ile, Centripetal metodun ortalama ve RMS hatası ile maksimum sapma miktarı düşürülmüştür.

Şekil 4.39'daki dördüncü data set, Boeing çalışanı ETY Lee'nin çalışmasından alınmıştır [54]. Burada köşelere yakın yerlerde daha çok veri noktası bulunmaktadır.

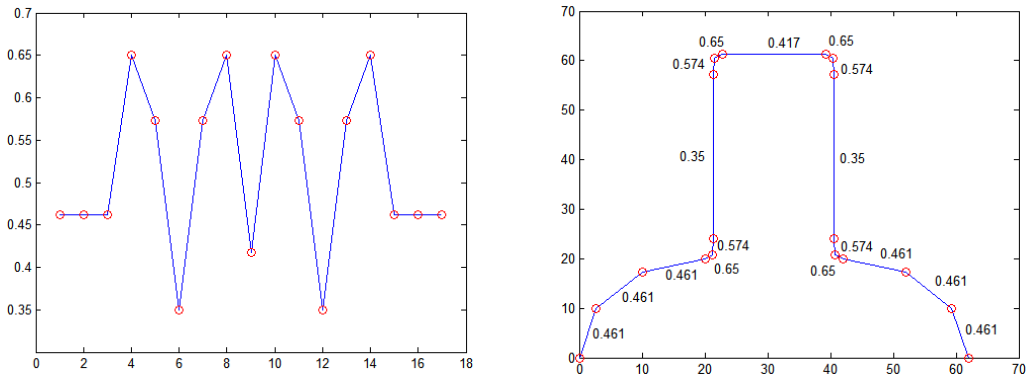


Şekil 4.39. Dördüncü data set ile kübik eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot



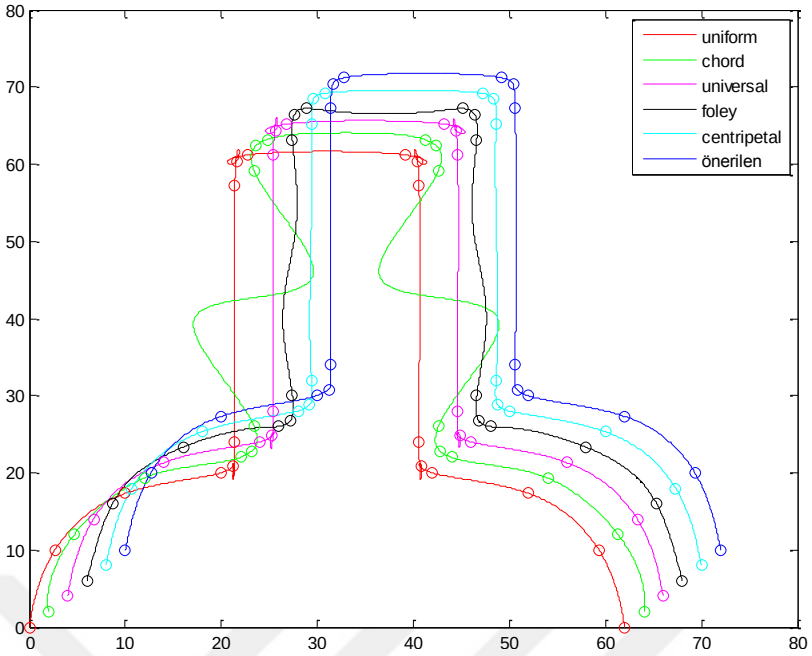
Şekil 4.39. (Devam) Dördüncü data set ile kübik eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot

Şekil 4.40'da dördüncü data set için kullanılan e_i üs değerleri, kiriş sırasına göre ve kirişler üzerinde gösterilmiştir.



Şekil 4.40. Dördüncü data set için e_i üs değerleri

Şekil 4.41'de çeşitli metotlara ve önerilen metoda göre, dördüncü data set ile üretilen eğriler gösteriliyor.



Şekil 4.41. Dördüncü data set için metotlara göre eğriler

Tablo 4.4. Şekil 4.39'daki eğri segmenti boyunca kirişten sapmalar

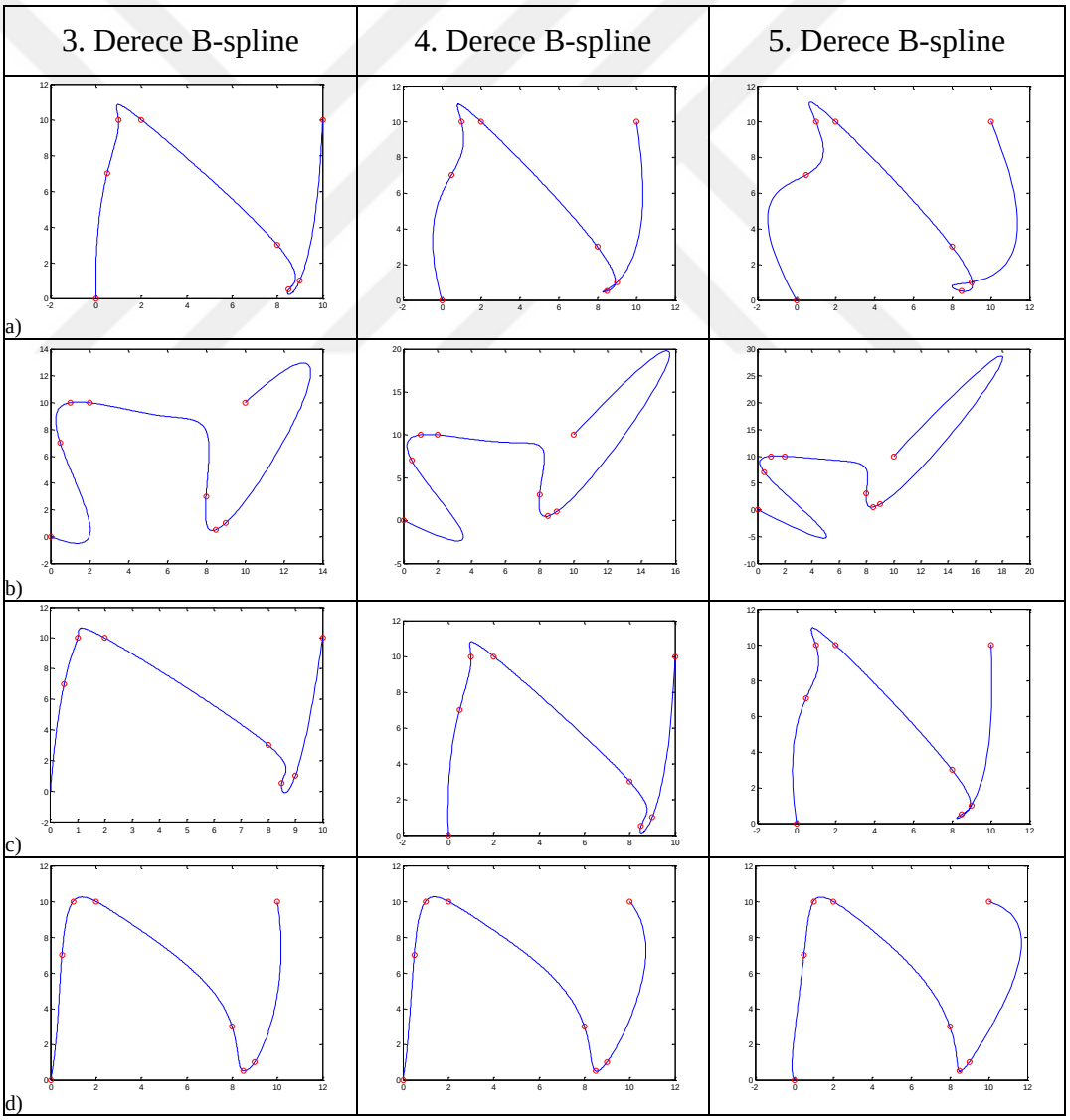
Metot	Uniform	Chord	Universal	Foley	Centripetal	Önerilen
1	0,4593	0,8607	0,4602	0,2704	0,3944	0,3685
2	0,9257	0,6698	0,9127	1,011	0,9574	0,9965
3	0,225	0,2848	0,2747	0,5958	0,2614	0,1802
4	0,109	0,8897	0,1558	0,1798	0,1449	0,0820
5	0,1584	0,4455	0,1461	0,1909	0,1756	0,1794
6	0,1879	5,224	0,2024	0,999	0,2223	0,1116
7	0,0998	0,4396	0,1009	0,1437	0,1264	0,0649
8	0,3779	0,084	0,3907	0,2422	0,1928	0,1859
9	0,4538	0,8721	0,4598	0,6513	0,3511	0,5599
10	0,3885	0,08393	0,3903	0,2422	0,1928	0,1859
11	0,06689	0,4396	0,092	0,1441	0,1278	0,0666
12	0,1419	5,224	0,2029	0,9916	0,2105	0,1313
13	0,1518	0,4433	0,1539	0,1937	0,1756	0,1794
14	0,1074	0,07757	0,1617	0,1655	0,1449	0,0823
15	0,2171	0,2919	0,2754	0,5958	0,2614	0,1797
16	0,9257	0,6717	0,9088	1,011	0,9574	0,9965
17	0,457	0,8607	0,4524	0,2782	0,3944	0,3798
Ortalama	**0,2830**	1,0508	0,3377	0,4651	0,2718	0,2900
RMS	0,4111	1,8705	0,4148	0,5741	0,3991	0,4080
Maks. Sapma	0,9257	5,224	**0,9127**	1,011	0,9574	0,9965
Süre (ms)	44,255	**43,387**	127,132	69,769	**43,429**	48,376

Tablo 4.4'de dördüncü data set için üretilen eğrilerin, kirişten maksimum sapmaları gösterilmiştir. Burada önerilen metot, centripetal metoda yakın bir başarım göstermiştir. Önerilen metot ile, centripetal metoda göre, uzun kirişlerde eğri kirişe daha iyi yaklaşmıştır.

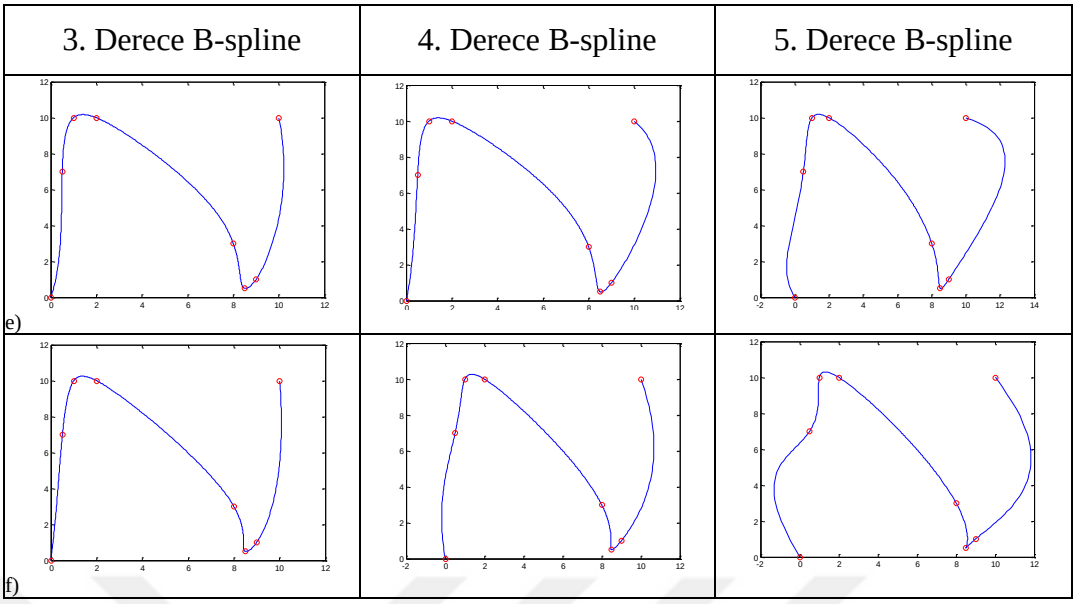
İlk üç data set için, önerilen metodun centripetal-merkezcil metoda göre iyileştirme sağladığı ve çoğu zaman Foley metodundan güçlü olduğu görülmektedir. Dördüncü data set için ise, tepelerdeki sapmaların ortalaması, centripetal metoda göre biraz kötü gibi görünse de, daha düzgün görünümlü bir eğri üretilmiştir. Bu durum, centripetal metoda göre, hata fonksiyonunun altında kalan alanın küçülmesi şeklinde gözlenmektedir.

4.8. Polinom Derecelerine Göre Parametrelendirme Başarımı

Şekil 4.42'de birinci data set için üretilmiş, 3., 4., ve 5. derece b-spline eğriler gösterilmiştir.



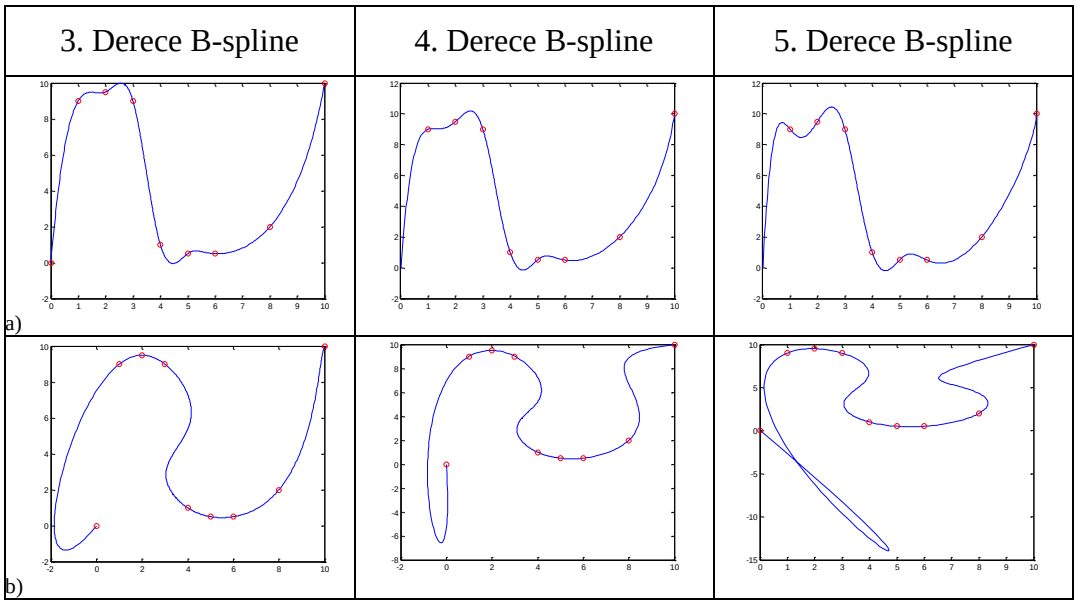
Şekil 4.42. Birinci data set ile eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot



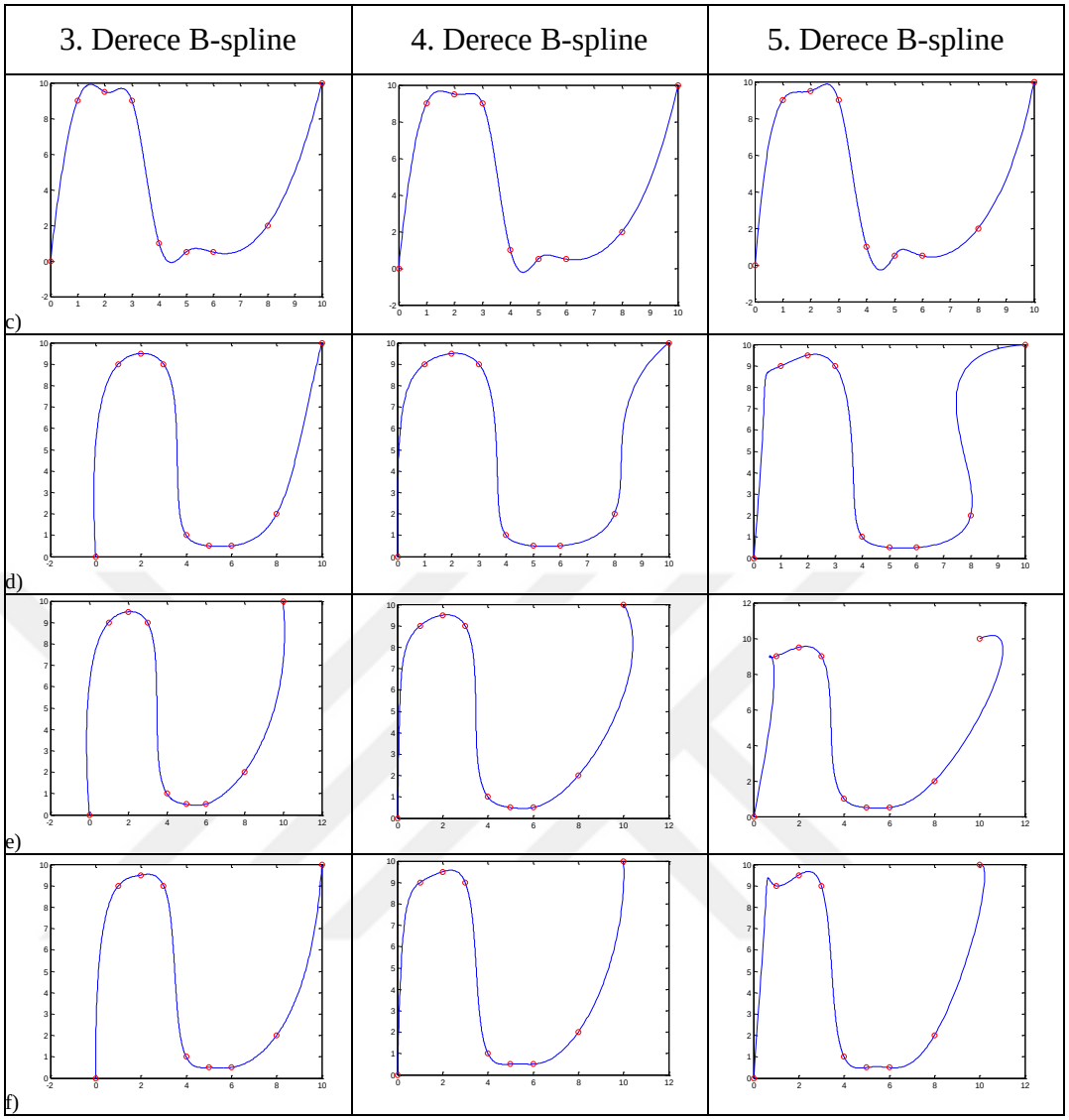
Şekil 4.42. (Devam) Birinci data set ile eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot

Birinci data set için, önerilen metot, centripetal metoda göre, 3. ve 4. derece eğrilerde salınımları azaltsa da, 5. derece eğride gereksiz salınımlar üretmiştir.

Şekil 4.43'de ikinci data set için 3. , 4. , ve 5. derece b-spline eğriler gösteriliyor. İkinci data set için, önerilen metot, centripetal metoda göre, 3. ve 4. ve 5.derece eğrilerde salınımları azaltsa da, 5. derece eğrideki gereksiz kıvrılma her iki metodda sürmektedir.



Şekil 4.43. İkinci data set ile eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot

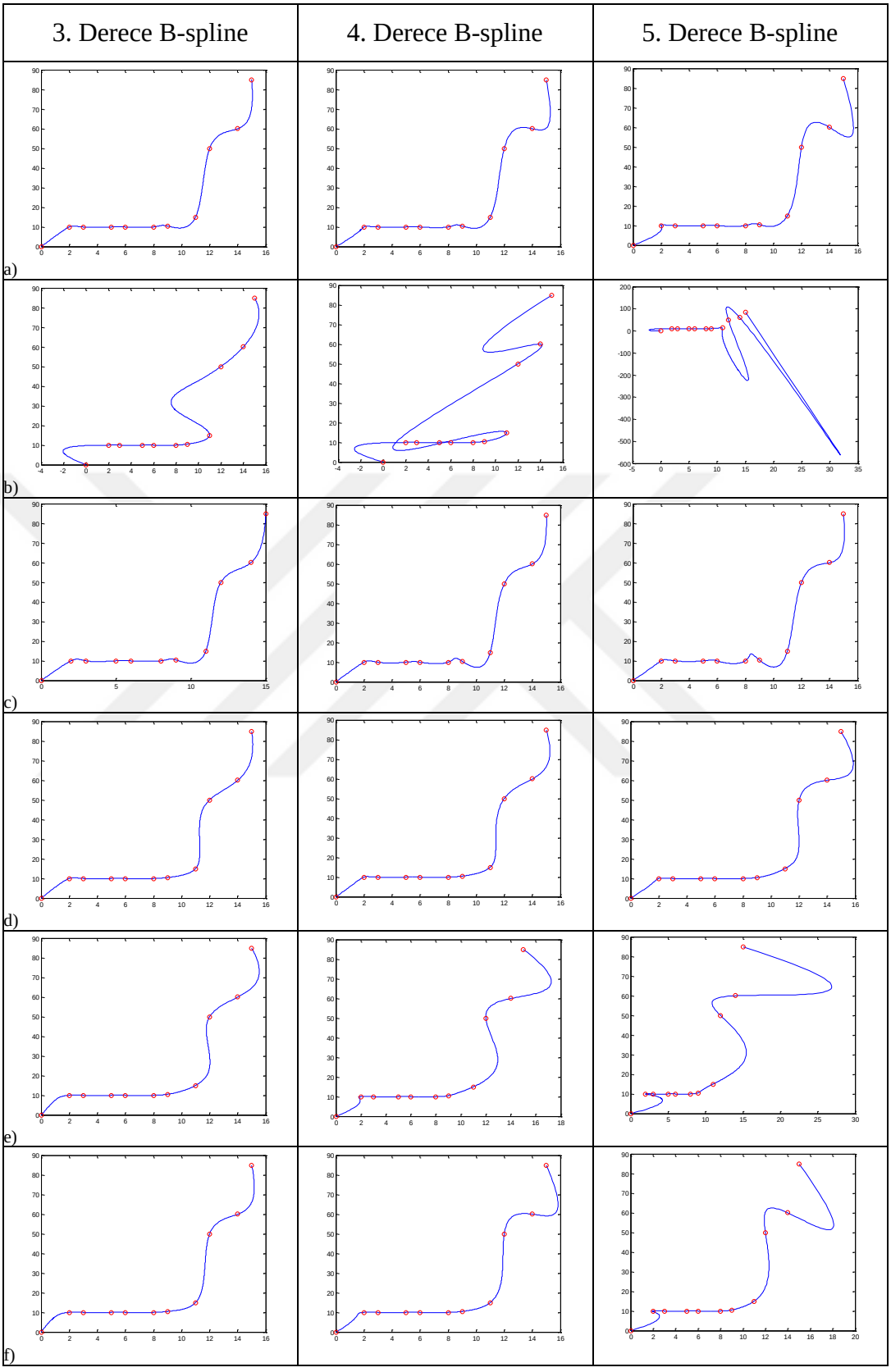


Şekil 4.43. (Devam) İkinci data set ile eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot

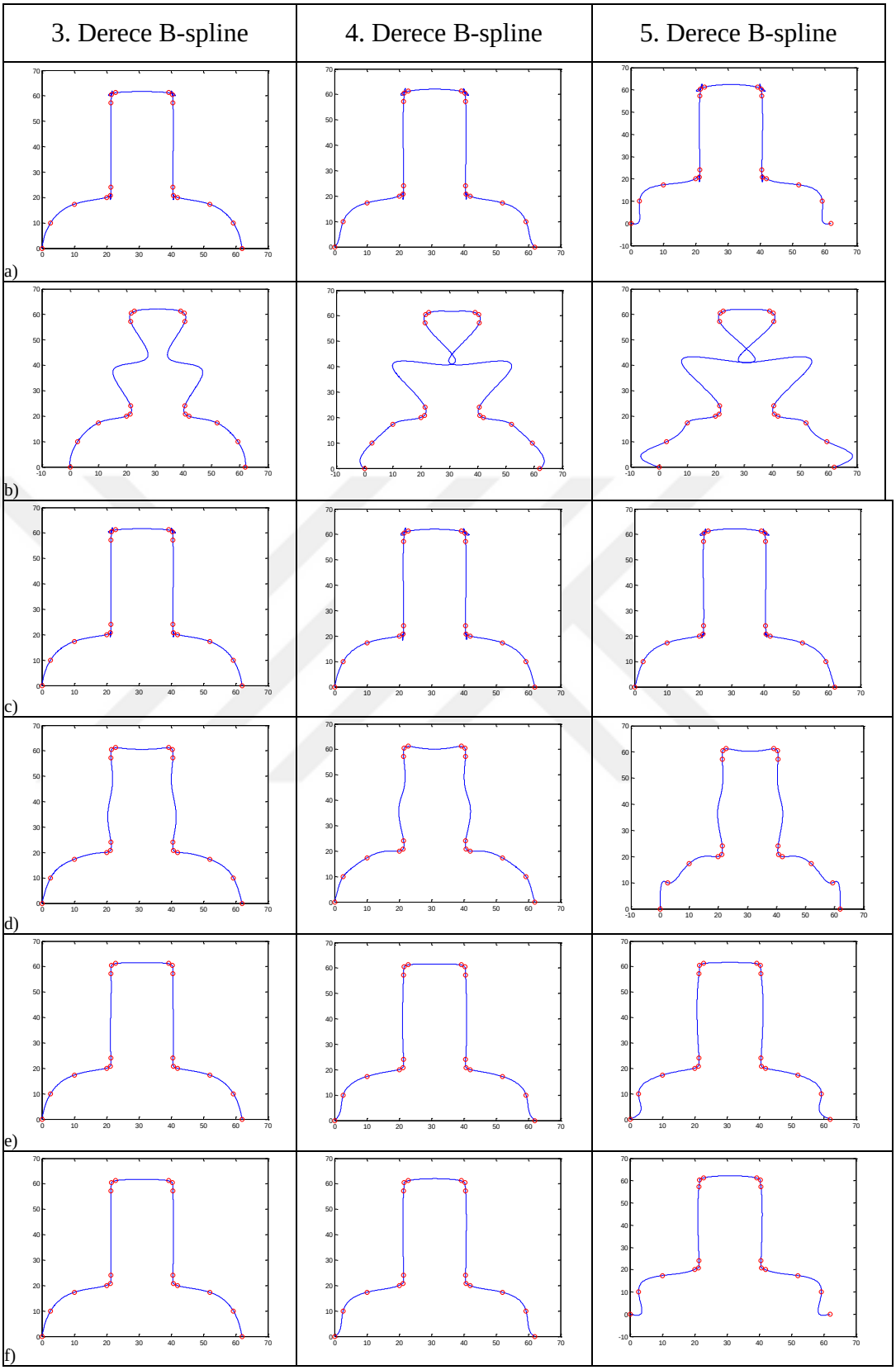
Şekil 4.44'de üçüncü data set için 3. , 4. , ve 5. derece b-spline eğriler gösteriliyor Üçüncü data set için, önerilen metot, centripetal metoda göre, 3. ve 4. ve 5.derece eğrilerde salınımları azaltmaktadır.

Şekil 4.45'de dördüncü data set için 3., 4., ve 5. derece b-spline eğriler gösterilmiştir.

Dördüncü data set için, önerilen metot, centripetal metoda göre, 3. ve 4. ve 5.derece eğrilerde salınımları azaltsa da, 5. derece eğride, uç noktada gereksiz kıvrılma üretmiştir.



Şekil 4.44. Üçüncü data set ile eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot



Şekil 4.45. Dördüncü data set ile eğri enterpolasyonu: (a). uniform (b). chord (c). universal (d). Foley (e). centripetal (f). önerilen metot

4.9. B-Spline Eğrilerin Türevleri

Kaskad geometri mimarisinde, temel geometrik denklemler ile ifade edilemeyen, serbest topolojiye sahip yüzeyler b-spline fonksiyonlar ile ifade edilmektedir. Bu yüzeylerin takım tezgahlarında işlenebilmesi için, yüzeyi teşkil eden b-spline eğrilerin türevlerinin sürekli olması istenir. Bu şekilde, her bir tarama diliminde, kesici kafanın malzeme üzerinde düzgün bir şekilde ilerlemesi garanti altına alınmış olur.

B-spline eğri denklemi aşağıdaki denklemdeki şekilde verilmiş olsun:

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \tag{4.46}$$

B-spline'ı üreten baz fonksiyonların türevleri ise şu şekilde hesaplanır:

$$\frac{d}{du} N_{i,p}(u) = N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{4.47}$$

Türev ifadesi eğri denkleminde yerine yazıldığında ise şu ifade elde edilir:

$$\frac{d}{du} C(u) = C'(u) = \sum_{i=0}^{n-1} N_{i+1,p-1}(u) Q_i \tag{4.48}$$

Q_i 'ler şu şekilde tanımlanır:

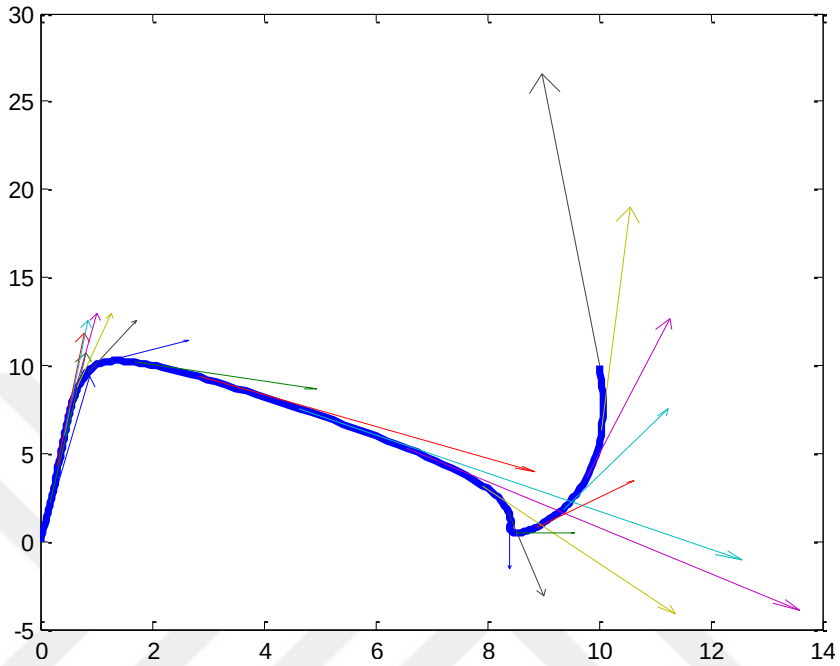
$$Q_i = \frac{p}{u_{i+p+1} - u_{i+1}} (P_{i+1} - P_i) \tag{4.49}$$

B-spline'ın türevi, orijinal düğüm vektörünü kullanan, $p-1$'inci dereceden, n adet $Q_0, Q_{1,\ldots} Q_{n-1}$ kontrol noktalarını ile üretilen yeni bir eğri şeklindedir.

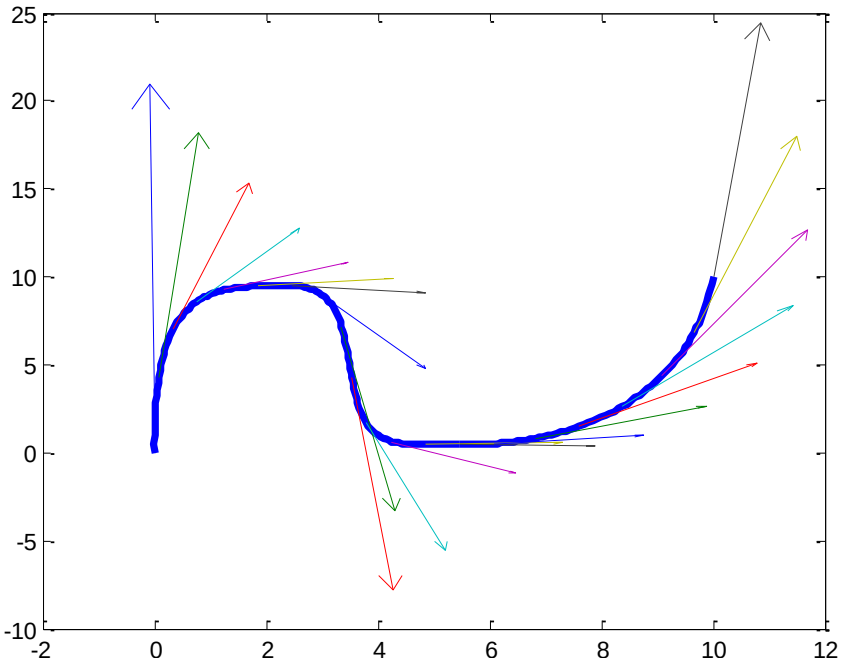
Clamped-sıkılmış b-spline eğrileri için ise, türev şu şekilde tanımlanır [52]:

$$\frac{d}{du} C(u) = C'(u) = \sum_{i=0}^{n-1} N_{i,p-1}(u) Q_i \tag{4.50}$$

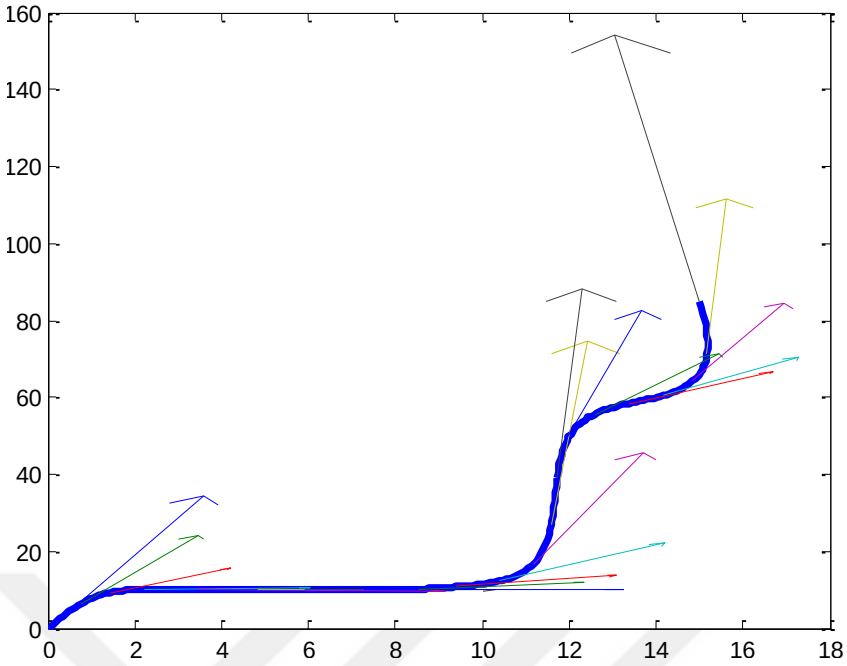
Şekil 4.46, 4.47, 4.48 ve 4.49'da, kullanılan veri setleri için, birinci türevler gösteriliyor. Bu çalışmada clamped – sıkılmış b-spline yapısı kullanıldığından, türevlemeler Denklem (4.50) kullanılarak elde edilmiştir.



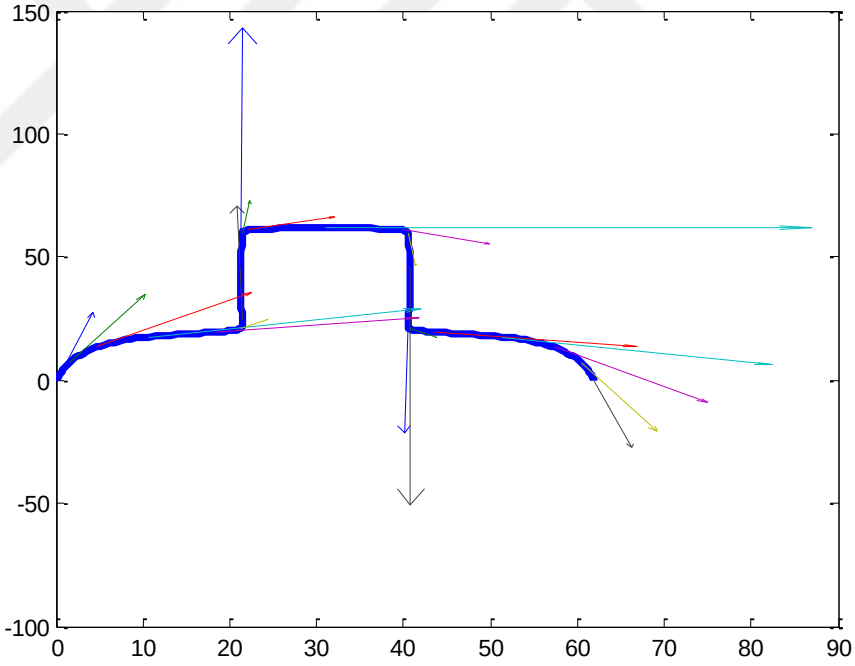
Şekil 4.46. Birinci data set için türevler



Şekil 4.47. İkinci data set için türevler



Şekil 4.48. Üçüncü data set için türevler



Şekil 4.49. Dördüncü data set için türevler

4.10. Yüzey İçin Parametrelendirme ve Düğüm Vektörü Üretimi

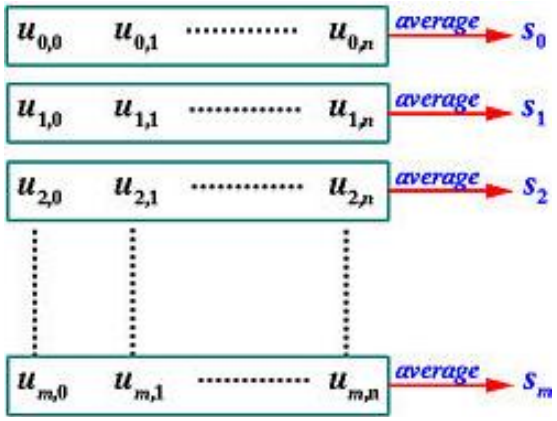
Grid data formatında verilen düzenli (x, y, z) veri kümesinin, b-spline yüzey enterpolasyon amaçlı olarak parametrelendirmesinde, eğri parametrelendirmeden farklı olarak, ilave bir ortalama alma aşaması bulunur. U yönündeki parametre ilerlemelerinin bulunmasından sonra, bu ilerlemelerin her bir adımdaki ortalaması,

yüzeyin *U* yönündeki parametre ve düğüm vektörüne temel oluşturur. Benzer şekilde *V* yönündeki paremetre ilerlemelerinin ortalamaları, *V* yönündeki düğüm vektörünü üretir. Yüzeyin *U* yönünü bir düğüm vektörü, *V* yönünü diğer bir düğüm vektörü temsil eder. Bir düğüm vektörü, aynı yöndeki birden çok eğriyi üretmiş olur.

B-spline yüzeyin derecesi (p, q) olarak tanımlanmış olsun. Kontrol noktaları $e+1$ satır ve $f+1$ sütuna dağılmış olsun. Denklem (4.51)'de B-spline yüzey ifadesi verilmiştir.

$$S(u,v) = \sum_{i=0}^{e} \sum_{j=0}^{f} N_{i,p}(u) N_{j,q}(v) P_{i,j} \tag{4.51}$$

Data noktaları da benzer şekilde $m+1$ satır ve $n+1$sütun olmak üzere tanımlanır. Data noktaları D_{ij} için, $0 <= i <= m$ ve $0 <= j <= n$ aralığındadır. *u* yönünde $m+1$ adet parametre $s_0,\ldots,s_m$ olarak tanımlanır. Benzer şekilde, *v* yönünde $n+1$ adet $t_0,\ldots,t_n$ parametreleri tanımlanır. *U* yönündeki parametrelerin hesaplanması için bir satırdaki bütün parametrelerin ortalaması alınır: $s_i = (u_{i,0} + u_{i,1} + \ldots + u_{i,n}) / (n+1)$. Şekil 4.50'de *U* yönündeki parametre hesabı gösterilmiştir.



Şekil 4.50. *u* yönünde parametre hesabı

Aynı şekilde *V* yönündeki parametrelerin hesaplanması için bir sütundaki bütün parametrelerin ortalaması alınır:

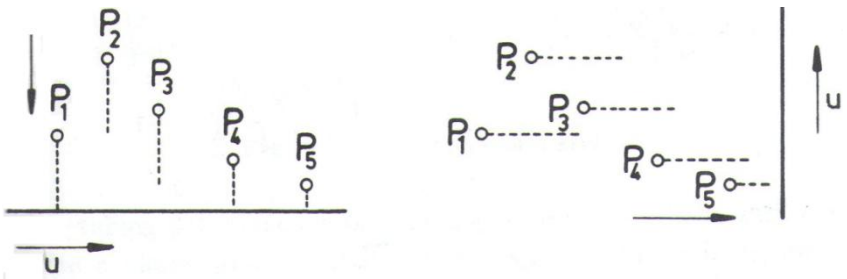
$$t_j = \frac{v_{0,j} + v_{1,j} + \ldots + v_{m,j}}{m+1}$$

u yönündeki $s_0, s_1 \ldots, s_m$ parametreleri ve *u* yönündeki *p* derecesi *U* düğüm vektörünü belirler. Aynı şekilde, *v* yönündeki $t_0, t_1 \ldots, t_n$ parametreleri ve *v* yönündeki *q* derecesi *V* düğüm vektörünü belirler [52].

4.11. Mesh Yüzeyden Grid Data Üretimi ve Yüzey Enterpolasyonu Uygulaması – IGES Formatına Geçiş

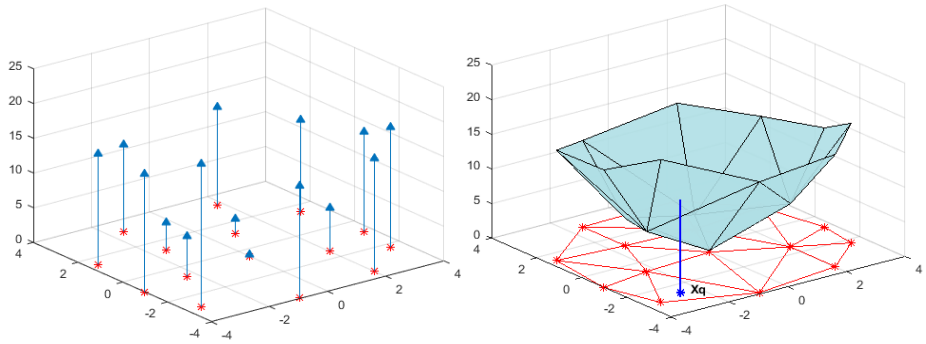
Mesh yüzeyden tek parçalı b-spline yama üretilmesi aşamasında, mesh yüzeyin parametrelendirilip grid data formatına aktarılmasında Peyre'nin [60] çalışmasındaki Floater [61] parametrelendirmesi ve grid data enterpolasyonu kullanılmıştır. Grid data üzerinden, b-spline enterpolasyon metodu ile tek parçalı b-spline yama üretilmiş ve bu yama IGES formatına dönüştürülerek CAD-CAM yazılımları ile işlenebilir hale getirilmiştir.

Mesh yüzeyin, grid data enterpolasyonuna girebilmesi için, mesh yüzeyin üçgen elemanlarının iki boyutlu bir düzlemde sıralanması gerekir. Bu şekilde *(u,v)* düzleminde, üçgen elemanların, soldan sağa ve yukarıdan aşağıya, mantıklı bir süreklilik üretecek şekilde, bire bir örten bir fonksiyonu sağlaması zorlanır. Şekil 4.51'de Hoschek'in kitabında [50], iki boyutlu data için bijective süreklilik anlatımı gösteriliyor.



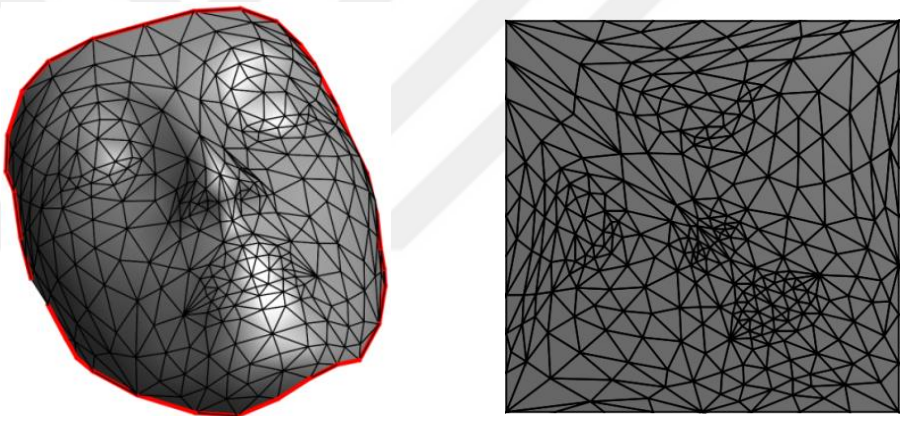
Şekil 4.51. Parametrelendirme izdüşüm yönü, 1-1 örten (bijective) [50]

Dağınık - scattered data noktalarının grid data formatına dönüştürülmesi için, öncelikli olarak komşuluk ilişkilerinin ortaya çıkarılması ve mesh yüzey elde edilmesi gerekir. Şekil 4.52'de Matlab Delaunay üçgenleme metodu ile veri noktalarından elde edilen üçgen yüzeyler gösteriliyor. Üçgenleme sonrasında, seçilen uygun bir düzlemden üçgen yüzeylere çıkarılan düzenli aralıktaki ışınların üçgenleri kestiği noktalar grid data için referans olarak alınır [62].

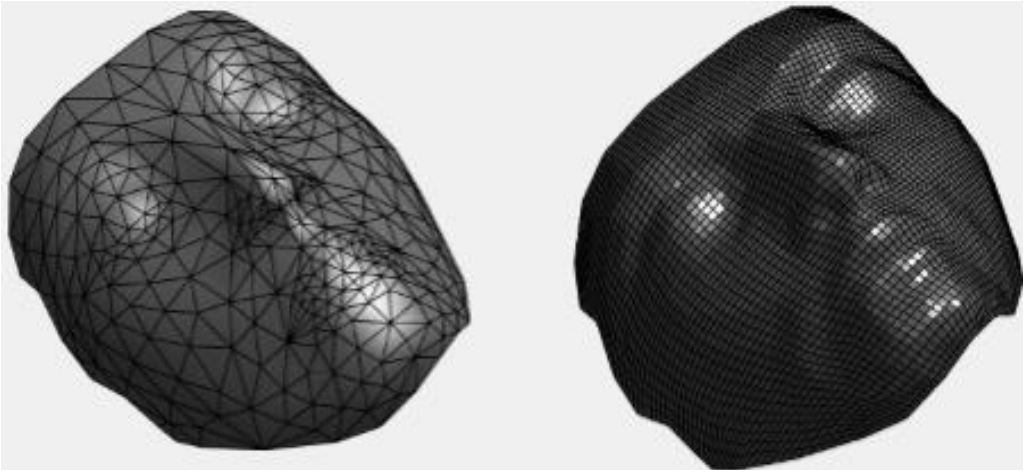


Şekil 4.52. Matlab düzensiz veri, lineer enterpolasyon

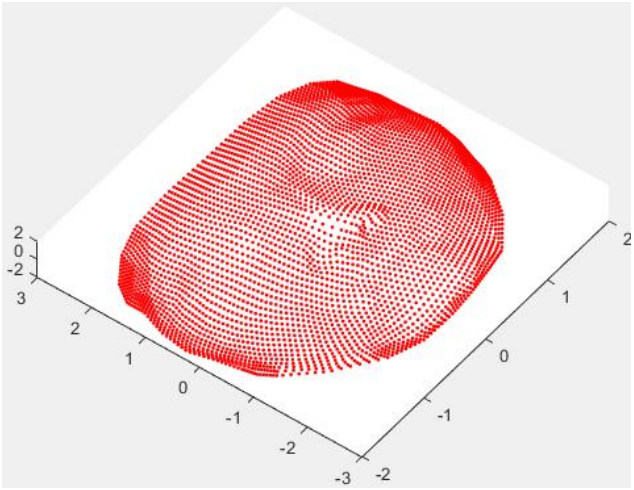
Bu çalışmada Peyre'nin [60] kütüphanesindeki Floater parametrelendirme ve grid data üretme fonksiyonları kullanılarak mesh yüzeyden grid data formatında veri üretilmiş, sonrasında grid data üzerinde üç boyutlu b-spline yüzey enterpolasyonu yapılmıştır. Şekil 4.53, 4.54 ve 4.55'de bu aşamalar gösteriliyor.



Şekil 4.53. Mesh parametrelendirme neticesi, Peyre/Floater [60, 61]

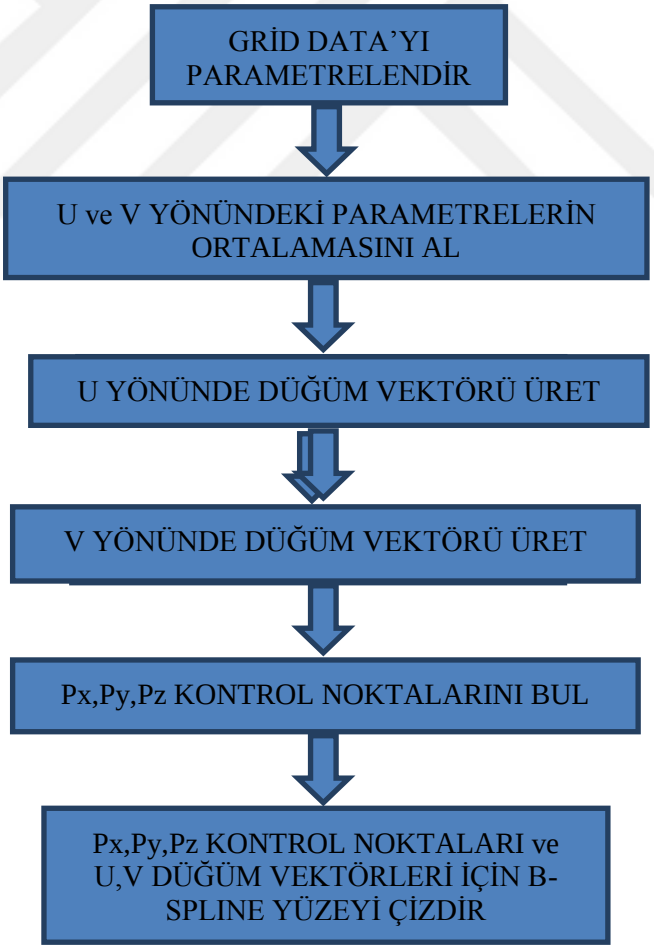


Şekil 4.54. Mesh yüzey ve elde edilen grid data



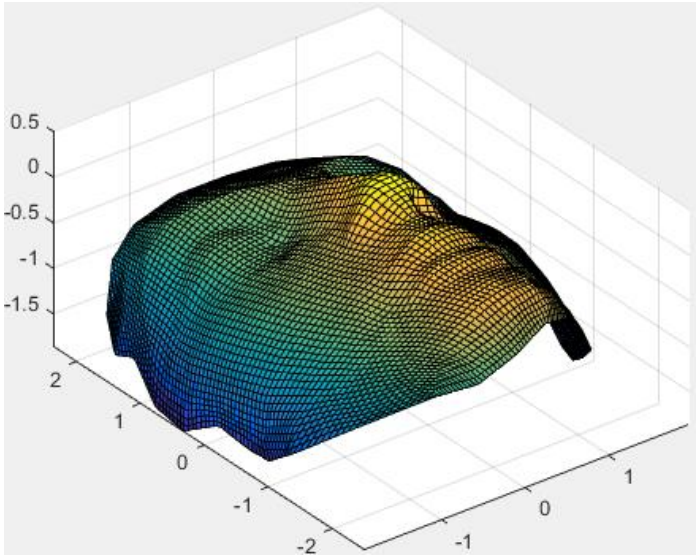
Şekil 4.55. Grid data noktalarının ayrık gösterimi

Grid datanın elde edilmesinden sonra, şekil 4.56'daki akış şemasındaki adımlar uygulanarak b-spline yüzey hesaplanır.



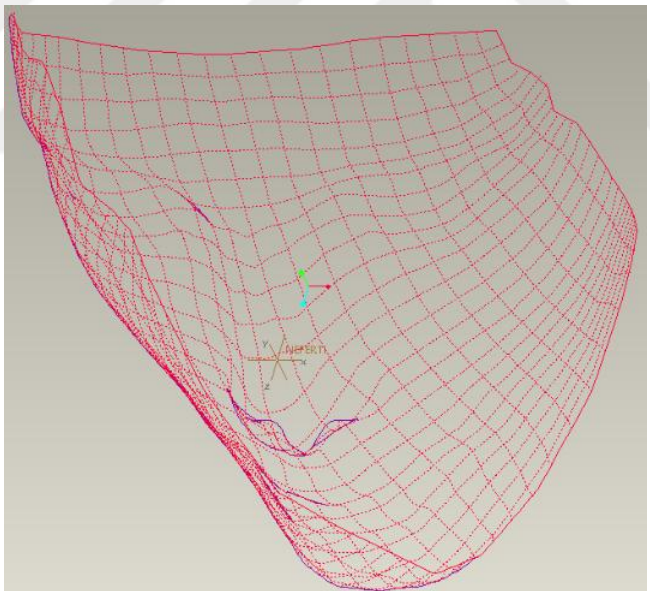
Şekil 4.56. Grid data – b-spline geçiş aşamaları

B-spline yüzeyin Nurbs toolbox formatında gösterimi şekil 4.57'de verilmiştir.



Şekil 4.57. B-spline yüzey

Şekil 4.58'de elde edilen b-spline yüzeyin IGES formatına dönüştürülmesi neticesi, Pro Engineer programı içindeki görüntüsü verilmiştir.

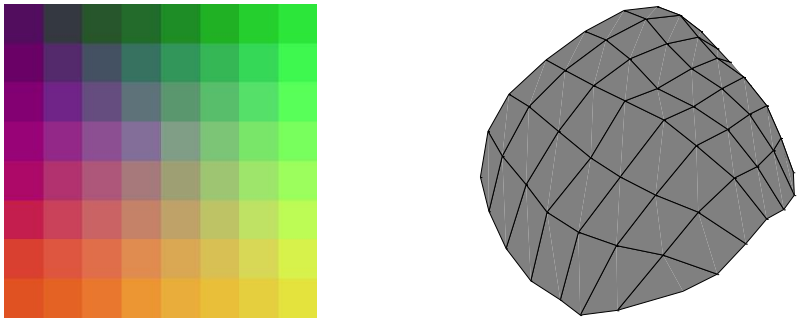


Şekil 4.58. IGES formatlı b-spline yüzey, pro engineer gösterimi

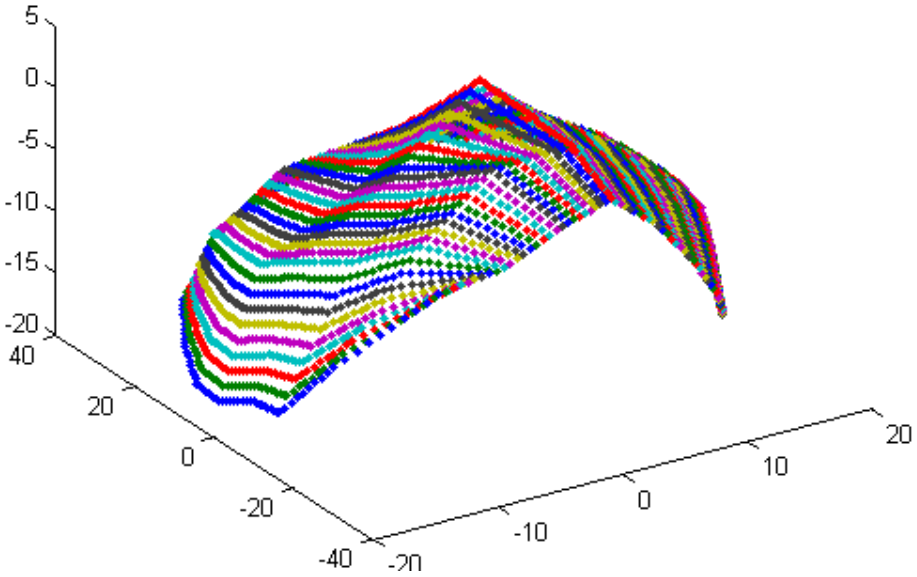
4.12. 3D Hata Analizleri

3D hata analizi için mesh üzerinden 8x8 boyutunda sadeleştirilmiş grid data üretilerek, centripetal – merkezcil ve önerilen dinamik merkezcil metotlarına göre 3D parametrelendirme ve yüzey enterpolasyonu yapılmıştır. 8x8 boyutundaki girdi grid data, *u* ve *v* yönlerinde 70x70 aralıkta lineer dağıtılarak, üretilen kesikli uzaydaki

noktaların b-spline yüzeye en yakın mesafeleri, hata ölçeği olarak alınmıştır. Şekil 4.59 ve 4.60'da girdi griz data ve 70x70 lineer dağıtılmış kesikli kiriş görülmektedir.



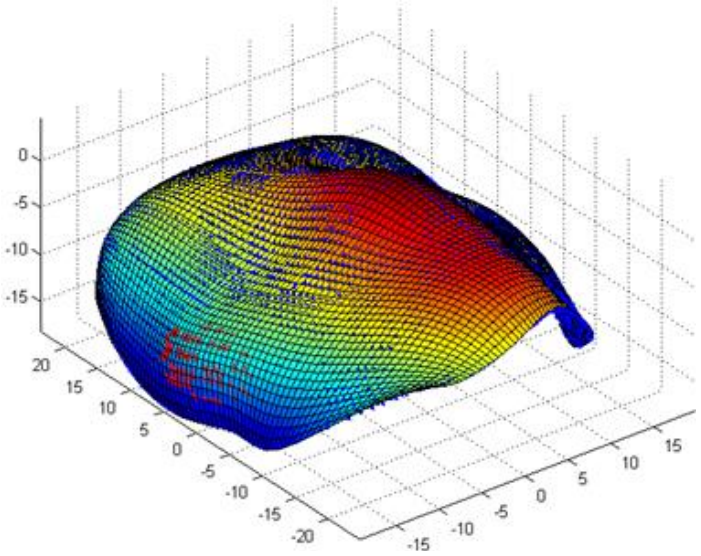
Şekil 4.59. 8x8 grid data



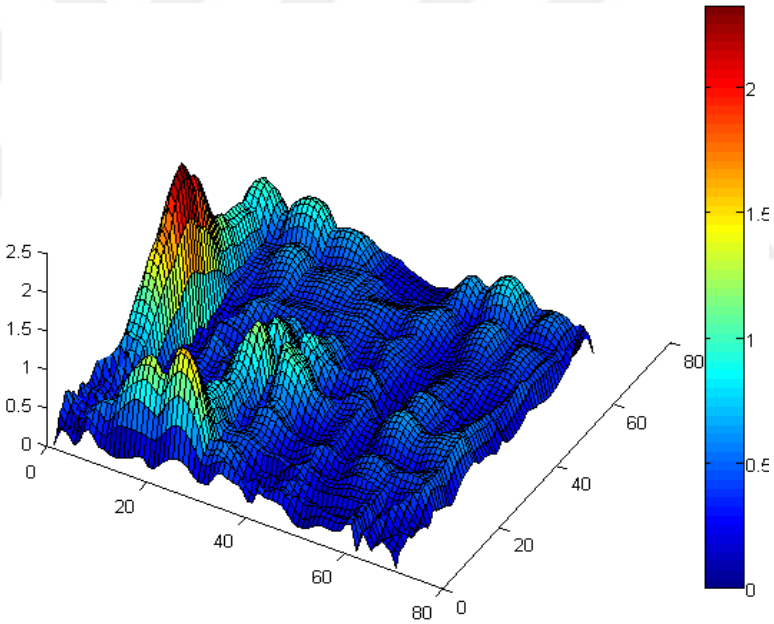
Şekil 4.60. 70x70 lineer dağıtılmış grid data

Centripetal–merkezcil parametrelendirmesi kullanılarak yapılan eğri enterpolasyonunda, eğrinin tepe noktalarının yüksek olduğu ve belirli bölgelerde yüksek sapmalar olduğu görülmüştür. Önerilen dinamik merkezcil parametrelendirme neticesinde, tepe noktalarında hata genliğinin düştüğü gözlenmiştir.

Şekil 4.61 ve 4.62'de centripetal metoda göre üretilen eğri, hata vektörleri ve hata grafiği gösteriliyor. Şekil 4.63 ve 4.64'de ise yeni önerilen dinamik centripetal metoda göre üretilen eğri, hata vektörleri ve hata grafiği gösteriliyor. Kırmızı renkli hata vektörleri, yüzeyin içine doğru yönlenmiş olan, boyu 1,5'den büyük hataları göstermektedir.

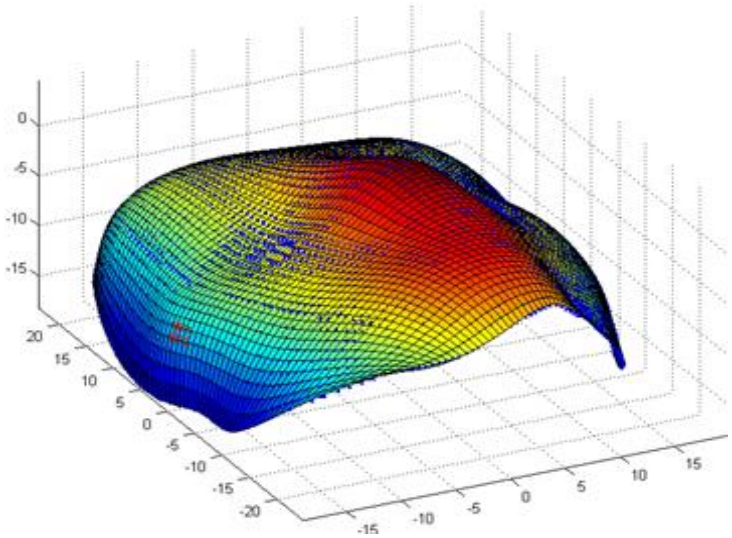


Şekil 4.61. Merkezcil metoda göre üretilen eğri ve hata vektörleri

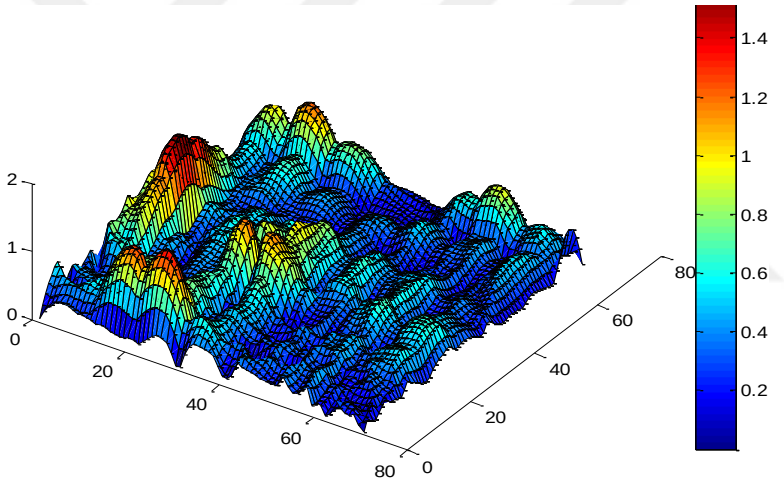


Şekil 4.62. Merkezcil metoda göre hata, toplam: 2318, ortalama: 0,4731

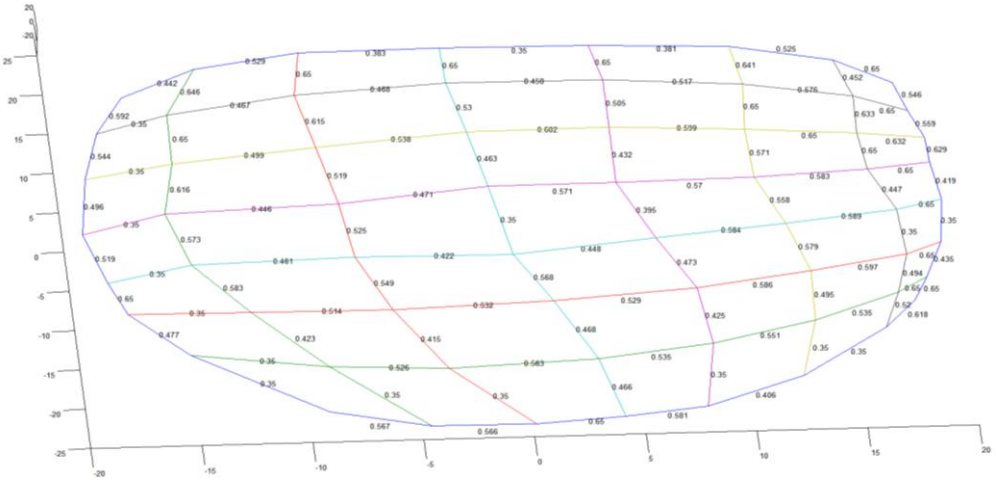
Şekil 4.65'de önerilen metod için kullanılan, *u* ve *v* yönlerinde hesaplanmış e_i dinamik üs değerleri, hedef noktaları birleştiren kirişler üzerinde gösteriliyor. e_i üs değeri, kiriş mesafesinin logaritması ile ters orantılı olarak dağıtıldığından, kısa mesafeli kirişlere yüksek üs değeri, uzun mesafeli kirişlere düşük üs değeri üretilmiştir. Böylece eğrinin uzun kiriş boyunca hızlanması sağlanmışıtır.



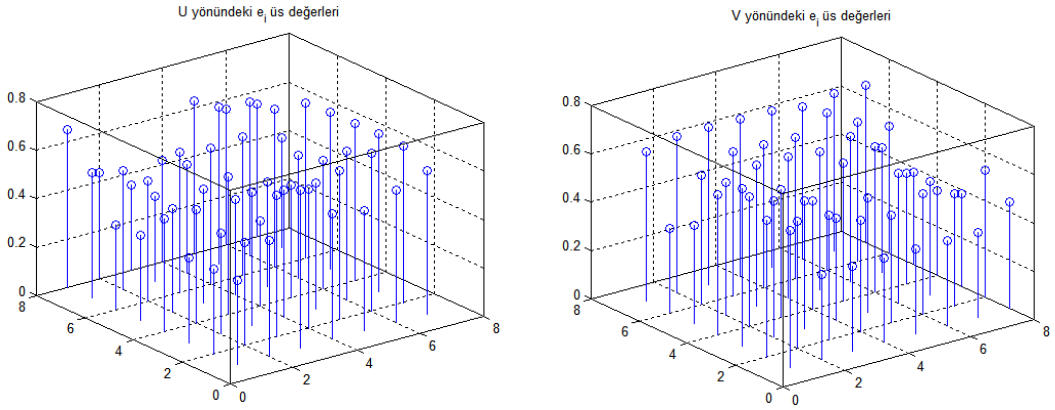
Şekil 4.63. Önerilen metoda göre üretilen eğri ve hata vektörleri



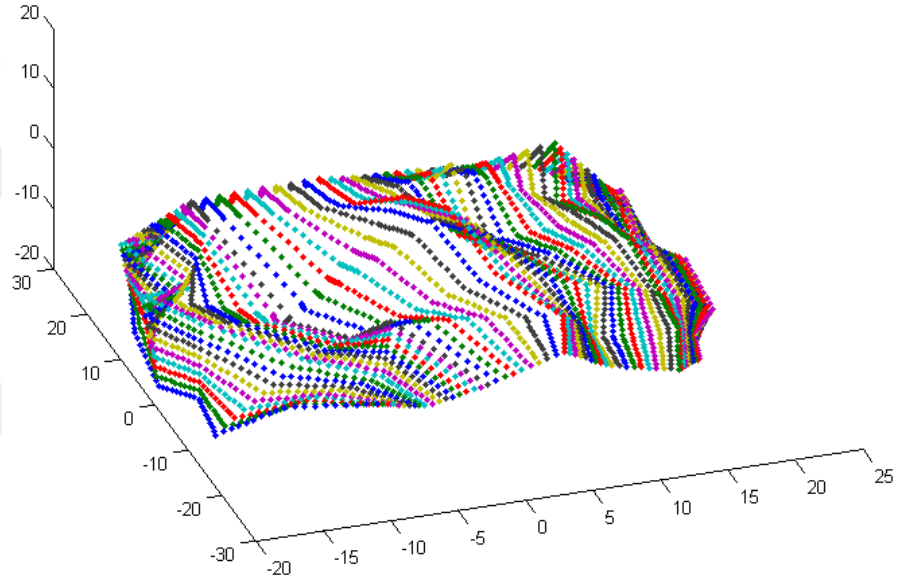
Şekil 4.64. Önerilen metoda göre hata, toplam: 2138, ortalama: 0,4365



Şekil 4.65. Önerilen metod, e_i dinamik üs değerleri

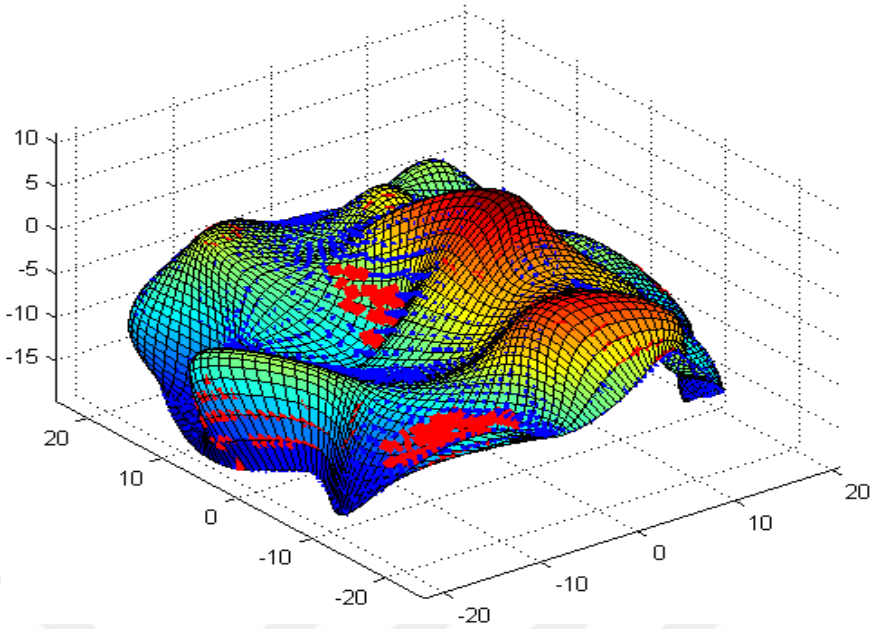


Şekil 4.66. Önerilen metod, e_i üs değerleri

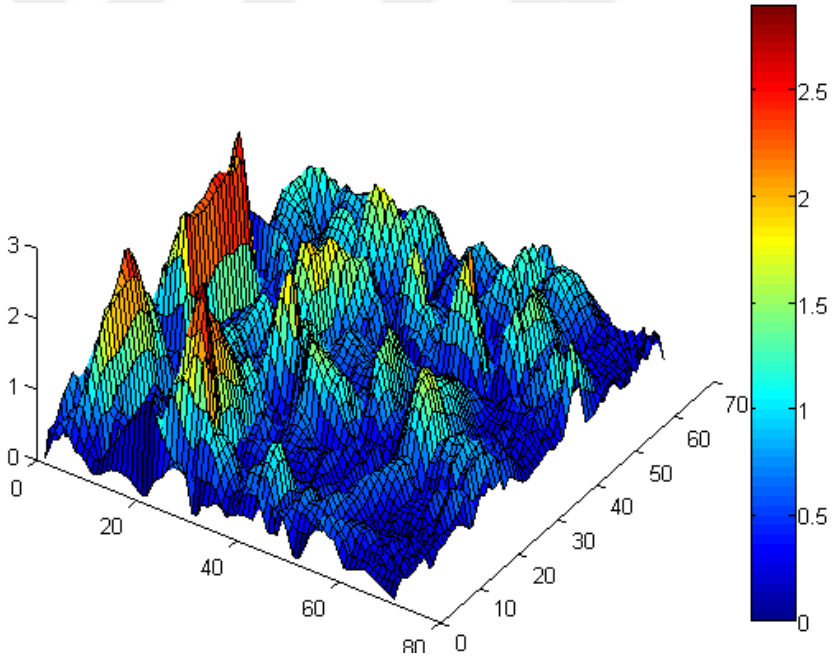


Şekil 4.67. 70x70 lineer dağıtılmış grid data, ikinci data set

Şekil 4.66'da ise, aynı veri, *u* ve v boyutları için, (*u, v*) kiriş uzayına göre ayrı ayrı gösteriliyor. Şekil 4.67'de keskin dönüşlere sahip ikinci data set gösteriliyor. İkinci data set için, şekil 4.68 ve 4.69'da centripetal metoda göre üretilen eğri, hata vektörleri ve hata grafiği gösteriliyor.

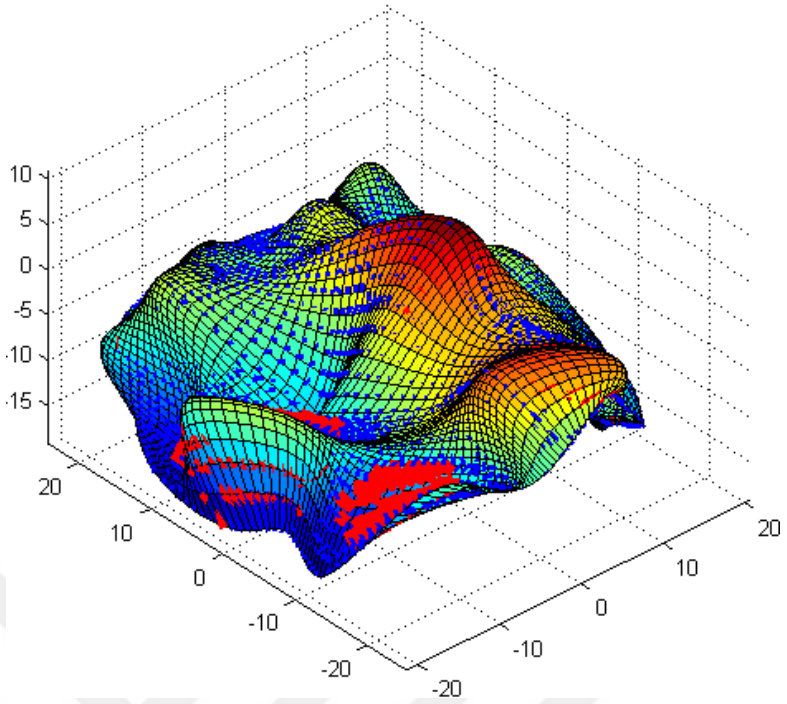


Şekil 4.68. Centripetal metoda göre üretilen eğri ve hata vektörleri, ikinci data set

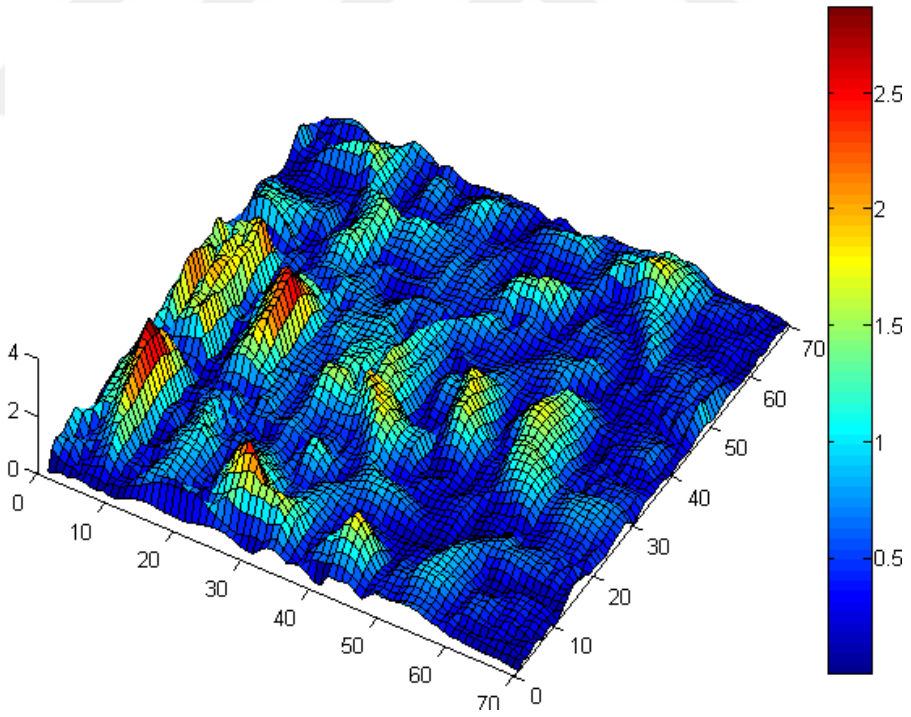


Şekil 4.69. Centripetal metod, ikinci data set için hata, toplam: 3170, ortalama: 0,647

Şekil 4.70 ve 4.71'de ise, ikinci data set için, önerilen dinamik centripetal metoda göre üretilen eğri, hata vektörleri ve hata grafiği gösteriliyor.

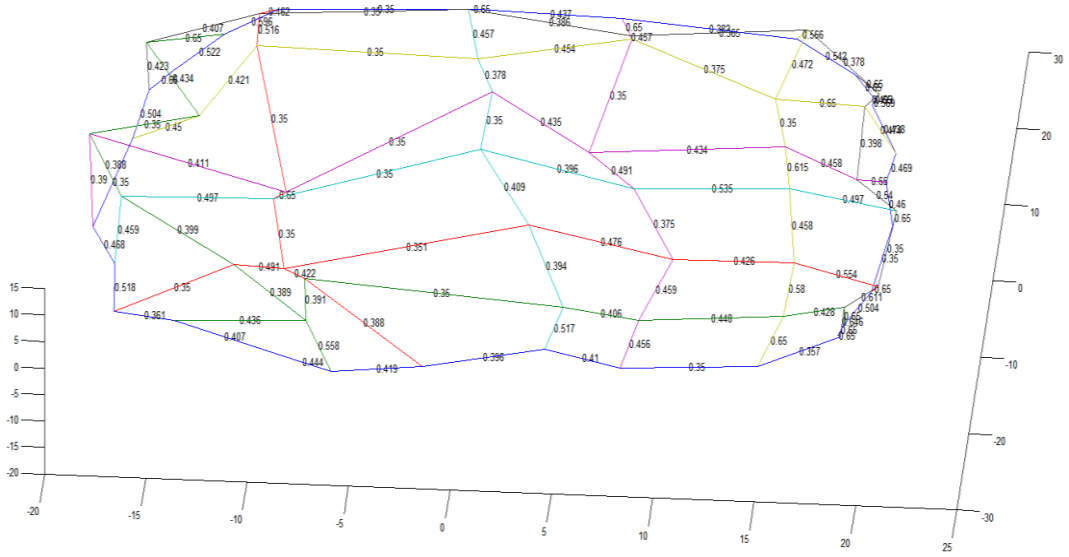


Şekil 4.70. Önerilen metoda göre üretilen eğri ve hata vektörleri, ikinci data set



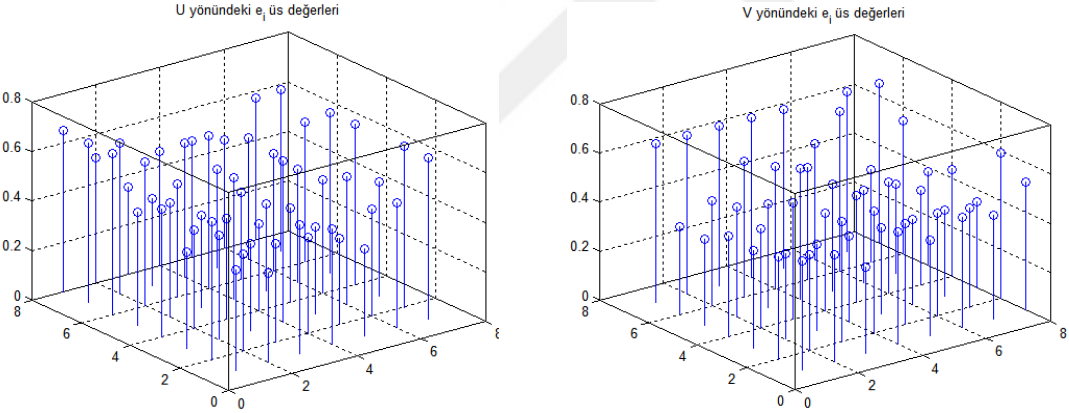
Şekil 4.71. Önerilen metod, ikinci data set için hata, toplam: 3343, ortalama: 0,682

Şekil 4.72'de önerilen metod için kullanılan, u ve v yönlerinde hesaplanmış e_i dinamik üs değerleri, hedef noktaları birleştiren kirişler üzerinde gösteriliyor.



Şekil 4.72. Önerilen metod, e_i dinamik üs değerleri

Şekil 4.73'de ise, aynı veri, *u* ve v boyutları için, (*u, v*) kiriş uzayına göre ayrı ayrı gösteriliyor.



Şekil 4.73. Önerilen metod, e_i üs değerleri

İkinci data set için, önerilen metod, verinin keskin dönüşlere sahip olmasından dolayı ve kiriş mesafelerinin düzenli dağılımından dolayı, ölçülen hataya iyileştirme sağlamamıştır. Önerilen metod, ani mesafe değişimlerinde, kiriş mesafelerine logaritmik etki ile uzaktan bakabildiğinden, değişken mesafeli kirişlere sahip, sürekli dış bükey yüzeylerde, güçlü bir iyileştirme etkisi üretebilmektedir.

5.SONUÇLAR VE ÖNERİLER

Bu tez kapsamında, bilgisayar destekli tasarım ve imalat, tıbbi görüntüleme ve benzeri alanlarda kullanılan, nokta bulutu verilerinden örme yüzey – mesh elde etme metotları, mesh sadeleştirme algoritmaları ve mesh yüzeye b-spline yama giydirmek üzere parametrelendirme metotları incelenmiştir. CAD yüzeyin ilk aşama modelini oluşturmak üzere b-spline yüzey enterpolasyon uygulamaları gerçekleştirilmiştir

Nokta bulutundan örme yüzeye geçiş aşamasında, tarayıcıdan gelen gürültülü veriye karşı dayanıklılık, nokta bulutundaki verilerin çokluğundan dolayı nokta sayısına göre üssel olarak artmayan zaman/işlem sayısı ve uygulamanın özelliğine göre keskin köşe bilgilerinin korunabilmesi gibi özellikler istenmektedir. Nokta bulutu işlemede radyal temelli fonksiyonlar kullanan algoritmalar, yoğun ve gürültülü veriler ile çalışabilmesinden dolayı son yıllarda öne çıkmaktadır [39].

Nokta bulutundan elde edilen örme yüzey, ilk aşamada gereğinden fazla üçgen poligonlar içerebilir. Topolojide kayıplar oluşturmadan bu fazlalıkların giderilmesi istenir. Bu durumda köşelerin özelliklerine göre sınıflandırılması ve belirli bir kritere göre elenmesi gerekir. Bu işlemler örme yüzeyin sadeleştirilmesi aşamasında yapılır.

Sadeleştirilmiş örme yüzeyin, anlamlı bir b-spline yama ile temsil edilmesi için ilk aşamada, yüzeyin bir düzleme izdüşümü alınarak, gölgeleme üretmeyen, mantıklı ilerleme yönlerinde iki boyutta parametrelendirilmesi gerekir. Bu parametreler, U ve V düğüm vektörlerini üretir. Üretilen düğüm vektörleri ve enterpolasyon veya yaklaşım neticesi bulunan kontrol noktaları, yüzeyi üreten b-spline yamanın girdi verilerini oluşturur.

Parametrelendirme aşamasında kullanılan çeşitli metotlar mevcuttur. Düzgün dağılımlı (uniform) parametrelendirme, data noktaları düzgün bir dağılıma sahip olduğunda işlevseldir. Ancak data noktaları düzensiz bir dağılıma sahip olduğunda verimli çalışmaz. Bu durumu aşmak üzere, parametreleri, data noktaları arasındaki

kiriş mesafesi ile orantılı olarak dağıtan kiriş mesafesi dağılımı (chord) kullanılmıştır. Ancak bu metod ani değişimlerde işlememektedir. Bu durumu düzeltmek üzere, E.Lee [54] tarafından, merkezcil parametrelendirme metodu (centripetal) önerilmiştir. Bu metoda, parametreler, kiriş mesafelerinin karekökleri ile orantılı olarak dağıtılmaktadır. Ayrıca kiriş geçişlerindeki açıları kullanan çeşitli metotlar önerilmiştir. Bölüm 4.7'de bu metotlar detaylı olarak açıklanmıştır.

Bu çalışmada, mevcut merkezcil - centripetal parametrelendirme metodunun iyileştirilmesi amaçlanmış ve bölüm 4.7.7'de ilave bir hesap metodu önerilmiştir. Mevcut CAD uygulamalarında, merkezcil metotta kiriş mesafelerinin kareköküne göre parametre dağılımı yapılmaktadır. Yaptığımız deneyler neticesi, bu üs değerinin kiriş mesafesinin logaritması ile ters orantılı olarak, belirli bir aralıkta dağıtılmasının, merkezcil metodun verimini arttırdığı gözlenmiştir. Önerilen dinamik merkezcil parametrelendirme metodu, parametre dağılımında her kiriş için sabit bir üs değeri kullanmak yerine, girdi veriye göre her kiriş mesafesi için ayrı hesaplanan dinamik üs değeri hesaplayarak karar vermektedir.

Tablo 4.1, 4.2, 4.3 ve 4.4'de, dört ayrı data set için, uniform, kiriş mesafesi, universal, Foley, merkezcil ve önerilen metotlar için parametrelendirme başarımları tablolar halinde gösterilmiştir. Önerilen metot, centripetal metoda göre, uzun kirişlerdeki aşırı eğri sapmalarını düşürmektedir.

Takip eden kısımda, aynı data setleri için, farklı parametrelendirme metotları kullanılarak, 3., 4. ve 5. derece polinomlardaki eğri davranışları incelenmiştir. Centripetal metoda göre, önerilen metot, eğrinin kirişten sapmalarını düşürmesine rağmen, 5. dereceki eğrilerde oluşan istenmeyen salınımlarda aynı davranışı göstermiştir.

Önerilen metotta kullanılan logaritma fonksiyonlarının, veri noktalarına zoom etkisinden dolayı davranışı lineer olmadığından dolayı, veri noktalarının verimli bir aralıkta normalize edilmesi gereklidir. Bu durum ileriki çalışmalarda göz önüne alınmalıdır.

Son aşamada, üç boyutlu bir örme yüzey üzerinde parametre üretilmesi, b-spline yama giydirilmesi ve IGES formatına dönüştürülmesi, uygulama olarak gösterilmiş,

mezkecil ve önerilen dinamik merkezcil metotlara göre 3D b-spline enterpolasyon hataları karşılaştırılmıştır. 3D hata analizlerinde ise, centripetal metod ve önerilen metod ile b-spline yüzey giydirilen iki veri seti için, hata grafikleri üretilmiştir. Değişken kiriş mesafelerine sahip olan ve sürekli dış bükey olan veri setinde, önerilen metodun sağlıklı bir iyileştirme etkisi ürettiği gözlenmiştir.

Bu çalışma, Kocaeli Üniversitesi Rektörlüğü Bilimsel Araştırma Projeleri BAP Koordinasyon Birimi Koordinatörlüğü tarafından 2015-018 no'lu proje kapsamında desteklenmiştir.

KAYNAKLAR

[1] Varady T., Martin R., Coxt J., Reverse Engineering of Geometric Models - an introduction, *Computer Aided Design*, 1997, **29**(4), 255-268

[2] Fabio R., From Point Cloud to Surface, *The Modeling and Visualization Problem,* Swiss Federal Institue of Technology, Zürih, 2003.

[3] Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W., Surface Reconstruction from Unorganized Points, *ACM Siggraph Computer Graphics*, 2002, **26**(2), 71-78.

[4] Lorensen W. E., Cline H. E., Marching Cubes: A high resolution 3D surface construction algorithm, *Computer Graphics*, 1987, **21**(4), 163-169.

[5] Rabbani T., Automatic Reconstruction of Industrial Installations Using Images and Point Clouds, PhD Thesis, University of Engineering and Technology, Lahore, Pakistan, 2005.

[6] Delaunay B., Sur la sphère vide, Izvestia Akademii Nauk SSSR, *Otdelenie Matematicheskikh i Estestvennykh Nauk*, 1934, **7**(1), 793-800.

[7] Voronoi G., Nouvelles applications des paramètres continus à la théorie des formes quadratiques, *Journal für die Reine und Angewandte Mathematik*, 1907, **133**(1), 97-178.

[8] Boissonnat D. J. D., Geometric structures for three-dimensional shaperepresentation, *ACM Transactions on Graphics*, 1984, **3**(4), 266-286.

[9] Schall O., Samozino M., Surface from Scattered Points, A Brief Survey of Recent Developments, *1st International Workshop on Semantic Virtual Environment*, Villars, Swiss, 16-18 March 2005.

[10] Amenta N., Bern M., Surface Reconstruction by voronoi filtering, *Discrete Computational Geometry*, 1999, **22**(1), 481-504.

[11] Amenta N., Choi S., Kolluri R., The power crust, *6th ACM Symposium on Solid Modeling*, Michigan, USA, 4-8 June 2001.

[12] Attene M., Spagnuolo M., Automatic surface reconstruction from pointsets in space, *Eurographics Computer Graphics Forum*, 2000, **19**(3), 457-465.

[13] Bernardini F., Mittleman J., Rushmeier H., Silva C., Taubin G., The ball-pivoting algorithm for surface reconstruction, *IEEE Transactions on Visualization and Computer Graphics*, 1999, **5**(4), 349-359.

[14] Bernardini F., Cohen-Steiner D., A greedy Delaunay based surface reconstruction algorithm, *INRIA*, 72024, 2002.

[15] Dey T. K., Goswami S., Tight cocone: A watertight surface reconstructor, *8th ACM Sympos. Solid Modeling Applications,* Seattle, WA, USD, 16-20 June 2003.

[16] Amenta H. N., Choi S., Dey T. K., Leekha N., A simple algorithm for homeomorphic surface reconstruction, *Internat. J. Comput. Geom. & Appl.*, 2002, **12**(1) , 125-141.

[17] Pulli K., Shapiro L.G.. Surface Reconstruction and Display from Range and Color Data. *Graphical Models,* 2000, **62**(1), 165–201.

[18] Cazals F., Giesen J., Yvinec M., Delaunay triangulation based surface reconstruction: a short survey, *INRIA,* 70609, 2004.

[19] Dey T. K., *Curve and surface reconstruction*, CRC Press, New York, 2004.

[20] Bajaj C. L., Bernardini F., Xu G., Automatic reconstruction of surfaces and scalar fields from 3D scans, *ACM SIGGRAPH 95*, Los Angeles, CA, USA, 6-11 August 1995.

[21] Carr J. C., Beatson R. K., Cherrie J. B., Mitchell T. J., Fright W. R., McCallum B. C., Evans T. R., Reconstruction and representation of 3D objects with radial basis functions, *ACM SIGGRAPH 2001*, Los Angeles, California , USA, 12-17 August 2001.

[22] Ohtake Y., Belyaev A., Alexa M., Turk G., Seidel H. P., Multi-level partition of unity implicits, *ACM Transactions on Graphics*, 2003, **22**(3), 463-470.

[23] Turk G., Dinh H. Q., O'Brien J., Yngve G., Implicit surfaces that interpolate, *Shape Modelling International*, Genova, Italy, 7-11 May 2001.

[24] Adamson A., Alexa M., Approximating and intersecting surfaces from points, *Symposium on Geometry Processing*, Aachen, Germany, 23-25 June 2003.

[25] Alexa M., Behr J., Cohen-Or D., Fleishman S., Levin D., Silva C. T., Computing and rendering point set surfaces, *IEEE Transactions on Visualization and Computer Graphics,* 2003, **9**(1), 3-15.

[26] Alexa M., Behr J., Cohen-Or D., Fleishman S., Silva C. T., Point set surfaces, *IEEE Visualization*, San Diego, California, USA, 21-26 November 2001.

[27] Amenta N., Kil Y. J., Defining point-set surfaces, *ACM Transactions on Graphics*, 2004, **23**(3), 264-270.

[28] Carr J. C., Beatson R. K., McCallum B. C., Fright W. R., McLennan T. J., Mitchell T. J., Reconstruction and representation of 3D objects with radial basis functions, *ACM GRAPHITE*, Melbourne, Victoria, Australia, 11-14 February 2003.

[29] Dey T. K., Goswami S., Provable surface reconstruction from noisy samples, *Computational Geometry*, 2006, **35**(1-2), 124-141.

[30] Kolluri R., Shewchuk J. R., O'Brien J. F.. Spectral surface reconstruction from noisy point clouds, *Symposium on Geometry Processing*, Nice, France, 8-10 July 2004.

[31] Ohtake Y., Belyaev A. G., Seidel H. P., 3D scattered data approximation with adaptive compactly supported radial basis functions, *Shape Modeling International* 2004, Cenova, Italy, 7-9 June 2004.

[32] Davis J., Marschner S. R., Garr M., Levoy M., Filling holes in complex surfaces using volumetric diffusion, *First International Symposium on 3D Data Processing, Visualization, and Transmission*, Padova, Italy, 19-21 June 2002.

[33] Ju T., Robust repair of polygonal models, *ACM Transactions on Graphics*, Los Angeles, California, 8-12 August 2004.

[34] Sharf A., Alexa M., Cohen-Or D., Contextbased surface completion, *ACM Transactions on Graphics*, Los Angeles, California, 8-12 August 2004.

[35] Verdera J., Caselles V., Bertalmio M., Sapiro G., Inpainting surface holes, *IEEE International Conference on Image Processing*, Barselona, Spain, 14-17 September 2003.

[36] Aim at Shape Proje Sunumu, Survey acquisition and reconstruction, *ACM Shape Modeling International*, Washington, DC, USA, 7-9 June 2004.

[37] Arya S., Mount D. M., Natanyahu N. S., Silverman R., Wu A. Y., An optimal algorithm for approximate nearest searching in fixed dimension, *Journal of the ACM*, 1998, **45**(6), 891-923.

[38] Shlen J. A tutorial on principle component analysis, 2005. http://www.cs.cmu.edu/~elaw/papers/pca.pdf (Ziyaret tarihi: 04 Kasım 2016)

[39] Wan L. C., *Surface Reconstruction By Layer Peeling*, PhD Thesis, School of Computing, National University of Singapore, 2007.

[40] Curless B., Levoy M., A volumetric Method for building complex modelsfrom Range Images, *23rd annual conference on Computer graphics and interactive techniques*, New Orleans, LA, USA, 4-9 August 1996.

[41] Güdükbay U., Çok Çözünürlüklü Modelleme için Poligonal Basitleştirme, *Sinyal İşleme ve Uygulamaları Kurultayı*, 14-15 Mayıs 1998.

[42] Jery O., Talton U., A Short Survey of Mesh Simplification Algorithms, *Cource Notes*, CS 598 MJG, 1-8, October 2004.

[43] Schroeder W. J., Zarge J. A., Lorensen W. E., Decimation of Triangle Meshes, *ACM Computer Graphics (SIGGRAPH'92 Proc.)*, **26**(2), 65-70, 1992.

[44] Knapp M., *Mesh Decimation Using VTK*, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Viyana, 2003.

[45] Garland M., Heckbert P. S., Surface Simplification Using Quadric Error Metrics, *ACM Computer Graphics*, 1996, **30**(2), 209-216.

[46] Solomon D., *Curves and Surfaces for Computer Graphics*, Springer, New York, 2006.

[47] David F. R., *An Introduction to NURBS*, Morgan Kaufmann Publishers, San Francisco, 2000.

[48] Piegl L., Tiller W., *The NURBS Book*, Springer, Germany, 1997.

[49] Song Y., 3D Free-form Surface Representation and Its Applications, PhD Thesis, School of Computer Science, University of Nottingham, 2007.

[50] Hoschek J., Lasser D., *Computer Aided Geometric Design*, A K Peter Ltd., Natick, MA, USA, 1993.

[51] Farin G., *Curves and Surfaces for CAGD: A Practical Guide*, Morgan Kaufman Publishers, San Francisco, CA, USA, 2002.

[52] Shene C. K., Introduction to Computing with Geometry, http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/ (Ziyaret tarihi: 04 Kasım 2016)

[53] Farouki R. T., Sakkalis T. J., Rational space curves are not "unit speed", *Computer Aided Geometric Design*, 2007, **24**(4), 238–240.

[54] Lee E., Choosing Nodes in Parametric Curve Interpolation, *Computer Aided Design*, 1989, **21**(6), 363-370.

[55] Lim C. G., A Universal Parameterization in B-spline Curve and Surface Interpolation, *Computer Aided Geometric Design*, 1999, **16**(5), 407-422.

[56] Foley T. A., Nielson G. M., Knot selection for parametric spline interpolation, *Mathematical methods in computer aided geometric design*, Academic Press Professional Inc., San Diego, CA, USA,1989.

[57] Fang, J. J., Hung, C. L., An improved parameterization method for B-spline curve and surface interpolation, *Elsevier Computer Aided Design*, 2013, **45**, 1005-1028.

[58] Irvine L., Marin S., Smith P., Constrained Interpolation and Smoothing, *Constructive Approximation*, 1986, **2**(1), 129-151.

[59] Fritsch F., Carlson R., Monotone Piecewise Cubic Interpolation, *SIAM Journal on Numerical Analysis*, 1980, **17**(2), 238-246.

[60] Peyre G., Numerical Tours, www.numerical-tours.com/ (Ziyaret tarihi: 04 Kasım 2016)

[61] Floater M. S., Floater Parametrelendirme, www.mn.uio.no/math/english/people/aca/michaelf/ (Ziyaret tarihi: 04 Kasım 2016)

[62] Matlab Dokumanları, Interpolating Scattered Data, www.mathworks.com/help/matlab/math/interpolating-scattered-data.html (Ziyaret tarihi: 04 Kasım 2016)

KİŞİSEL YAYIN VE ESERLER

Balta C., Oysu C., Bingül Z., Öztürk S., XYZ Kartezyen Robot ve 2B CAD CAM Çizici Yazılımı Tasarımı, *Otomatik Kontrol Ulusal Toplantısı (TOK'09)*, YTÜ, İstanbul, 13-16 Ekim 2009.

Balta C., Öztürk S., Oysu C., Görme Destekli Kartezyen Robot İçin Kenar Resmi Vektorizasyon Uygulaması, *13. Ulusal Kongre*, ODTÜ, Ankara, 23-26 Aralık 2009.

Balta C., Öztürk S., Tersine Mühendislikte Laser Tarama Nokta Bulutundan Örme Yüzey Elde Edilmesi, *13. Ulusal Kongre,* ODTÜ, Ankara, 23-26 Aralık 2009.

Balta C., Öztürk S., Bingül Z., RPR Robot Mekanizması İçin İleri Kinematik Hesaplarının Simulink Ortamında Parametrik Olarak Gerçeklenmesi, *M.M.O. Endüstriyel Otomasyon Sempozyumu ve Sergisi*, İTÜ, İstanbul, 21-22 Ekim 2011.

Balta C., Öztürk S., Tersine Mühendislik Uygulamalarında Nokta Bulutu Verilerinden Parametrik Yüzey Denklemleri Elde Etmede Aşamalar, *Otomatik Kontrol Ulusal Toplantısı (TOK 2014)*, KOÜ, Kocaeli, 11-13 Eylül 2014.

Öztürk S., **Balta C.,** Kuncan M., B-spline Eğri Enterpolasyonu İçin Kullanılan Parametrelendirme Metodlarının Karşılaştırılması, *International Engineering Science and Education Conference (INESEC)*, Diyarbakır, 1-3 Aralık 2016.

ÖZGEÇMİŞ

Kocaeli-İzmit doğumlu Cengiz Balta, lisans eğitimini burslu olarak Bilkent Üniversitesi Bilgisayar ve Enformatik Mühendisliği Bölümü'nde tamamladı. Bir süre özel sektörde çalıştıktan sonra, yüksek lisans eğitimini Kocaeli Üniversitesi Mekatronik Mühendisliği Bölümü'nde sürdürdü. Doktora eğitimi boyunca Kocaeli Üniversitesi Mühendislik Fakültesi'nde araştırma görevlisi olarak çalıştı.