YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SAYISAL İŞARET VE GÖRÜNTÜ İŞLEMEDE B-SPLİNE FONKSİYONLARI İLE İNTERPOLASYON VE FİLTRELEME

YÜKSEK LİSANS TEZİ

ELEKTRONİK MÜH. SONGÜL ALBAYRAK

İSTANBUL/1993

Yüksek Lisans öğrenimim boyunca bana her konuda yardımcı ve destek olan değerli hocam Sayın Prof. M.Yahya KARSLIGİL'e, bu projenin hazırlanmasında ve oluşmasında olanaklarından faydalandığım TÜBİTAK MAM. elemanlarından Sayın Derya DEMİR'e, tezimin yazılmasında emeği geçen tüm arkadaşlarıma teşekkür ederim.

Şubat-1993 Songül ALBAYRAK

İÇİNDEKİLER

ÖZET

Bu tez çalışması, çeşitli işaret işleme uygulamalarında bir araç olarak B-spline kullanımını tanıtır. Polinomla interpolasyon ve B-spline teorileri kısaca gözden geçirilerek B-spline interpolasyon ve filtreleme konuları incelenmiştir. Bilgisayar uygulamalarında etkili bir yazılım üzerinde durulmuştur.Sonuçta iki boyutlu görüntü formatında deneysel sonuçlar gösterilmiştir. Görüntü ve işaret işleme uygulamaları interpolasyon, düzgünleştirme büyültme ve küçültme gibi işlemlerdir.

Mühendislik analiz ve bakış açısıyla Cubic B-spline fonksiyonunun düzgünleştirme ve interpolasyon özelliği ile çalıştık. Fonksiyonun lokal destek ve ötelemeyle değişmeme özellikleri çok etkileyici işlemsel prosedürler sunar. Bilgisayar simülasyonu ile büyültme uygulamalarında Cubic B-spline'ın diğer interpolasyon metodlarından daha iyi olduğu gösterilmiştir.

ABSTRACT

This thesis presents the use of B-splines as a tool in various digital signal processing applications. The theory of polinomial interpolation and B-splines are briefly reviewed, followed by discussion on B-spline interpolation and B-spline filtering. Computer implementation using an efficient software viewpoint is discussed. Finally, experimental results are presented for illustrative purposes in two-dimensional image format. Applications to image and signal processing include interpolation, smoothing, filtering, enlargement and reduction .

From an engineering analysis and application point of view, we have studied the interpolation and smoothing property of the cubic B-spline function. Its local support and shift-invariant properties offer very attractive computational procedures. Computer simulations with applications to image magnification, minification enlargement have shown that the cubic B-spline is superior to other interpolation methods.

1.1 İNTERPOLASYON

Basit olarak interpolasyon işlemi tablo halinde değerleri verilen bir değişkenin tabloda olmayan bir değerini bulma işlemi olarak tanımlanabilir. Genel anlamda ise interpolasyon, bilinmeyen bir f(x) fonksiyonunun x_0, x_1, x_2, ... x_n gibi ayrık noktalarda verilen f_0, f_1, f_2 ..f_n değerlerini kullanarak, bu fonksiyonun daha basit ve bilinen bir F(x) fonksiyonu ile ifade edilmesi işlemidir. Bulunan F(x) fonksiyonuna 'interpolasyon fonksiyonu' denir ve polinom, trigonometrik fonksiyon, üslü ifade veya özel bir fonksiyon gibi çeşitli şekillerden birine sahip olabilir. Ama çoğu kez interpolasyon fonksiyonları olarak polinomlar kullanılırlar. Verilen değerlerin periyodik olması durumunda trigonometrik fonksiyonlar da kullanılabilir.

Bu bölümde interpolasyon fonksiyonu olarak sadece polinomlar incelenmiştir.

Sonlu farklar interpolasyon yöntemleri pratik olmalarına rağmen ayrık x_0, x_1, ... x_n noktaları arasındaki aralığın eşit olduğu durumlarda uygulanabilir.Ayrık noktalar arası eşit değilse, doğrusal interpolasyon, Lagrange interpolasyon yöntemi ve Aitken interpolasyon yöntemi kullanılabilir.

1.2 DOĞRUSAL İNTERPOLASYON

İnterpolasyon fonksiyonu olarak birinci dereceden bir polinom kullanılırsa doğrusal (lineer) interpolasyon adını alır.

Eğer x değişkeninin [a,b] aralığında bir f(x) fonksiyonu verilmiş ve interpolasyon fonksiyonu olarak

$$F(x)=Ax+B$$

kullanılacak ise

$$f(a)=F(a)$$
$$f(b)=F(b)$$

bağıntılarının sağlanması gerekir. Buradan da

$$A.a+B=f(a)$$
$$A.b+B=f(b)$$

yazılır ve bilinmeyen A,B katsayılarının çözümü

$$A=(f(b)-f(a))/(b-a)$$
$$B=(bf(a)-af(b))/(b-a)$$

verir.Böylece F(x) fonfsiyonu için

$$F(x)=(f(b)-f(a))/(b-a)x+(bf(a)-af(b))/(b-a)$$

elde edilir. $W_0(x)$ ve $W_1(x)$ gibi iki ağırlık fonksiyonunu

$$W_0(x)=(b-x)/(b-a)$$
$$W_1(x)=(x-a)/(b-a)$$

olarak tanımlayıp yukarıdaki denklemde yerine yazarsak

$$F(x)=W_0(x)f(a)+W_1(x)f(b)$$

olarak yazılabilir ve x değişkeni $a \leq x \leq b$ aralığında ise

$$W_0(x)+W_1(x)=1$$

bağıntısı sağlanır. Doğrusal interpolasyon için işlemdeki kesme hatası;

$$f(x)-F(x)=(x-a)(x-b)/2f''(c) \qquad a\leq c\leq b$$

ile verilir. Görüldüğü gibi eğer f(x) fonksiyonu da doğrusal ise f''(x) sıfır olacağından hata da sıfırdır. Diğer bir deyişle doğrusal fonksiyonlar doğrusal interpolasyon fonksiyonuyla tam olarak ifade edilebilirler.

1.3 LAGRANGE İNTERPOLASYON FORMÜLÜ

Bilinmeyen bir f(x) fonksiyonunun x_0, x_1,...x_n gibi (n+1) tane ayrık noktada bilinen değerlerini

$$f_0=f(x_0),\ f_1=f(x_1),\ .....\ f_n=f(x_n)$$

ile göstererek Lagrange interpolasyon formülü

$$F(x)=\sum_{i=0}^{n} L_i(x)\,f_i$$

şeklinde yazılır. Burada L_i terimleri

$$Li(x)=\prod_{\substack{j=0\\ j\neq i}}^{n}(x-x_j)/(x_i-x_j)$$

şeklinde tanımlanmıştır. Görüldüğü gibi her $L_i(x)$ katsayısı n dereceli bir polinomdur.
Lagrange yönteminin yararlı yanı, kullanılan ayrık noktaların eşit aralıklarla verilmesinin gerekmemesidir.

1.4 NEWTON İNTERPOLASYON FORMÜLLERİ

1.4.1 Newton İleriye Doğru Fark İnterpolasyon Formülleri

$$F(s)=f_0+\binom{s}{1}\Delta f_0+\binom{s}{2}\Delta^2 f_0+\ .\ .\ .\ +\binom{s}{n}\Delta^n f_0$$

Formülü Newton'un ileri doğru farklar interpolasyon formülüdür.Burada kullanılan boyutsuz s değişkeni $s=(x-x_0)/h$ olarak tanımlanmıştır. h fark tablosundaki ardışık iki x değeri arasındaki farkı gösterir.

i	x_i	$f(x_i)$	Δf_i	$\Delta^2 f_i$
1	x_0	$f(x_0)$		
			$\Delta f_0=f(x_1)-f(x_0)$	
2	x_1	$f(x_1)$		$\Delta^2 f_0= \Delta f_1-\Delta f_0$
			$\Delta f_1=f(x_2)-f(x_1)$	
3	x_2	$f(x_2)$		$\Delta^2 f_1= \Delta f_2-\Delta f_1$
			$\Delta f_2=f(x_3)-f(x_2)$	
4	x_3	$f(x_3)$		

Yukarıda verilen ileri fark tablosunu kullanarak interpolasyon polinomu ileri fark terimleri ile

$$F(x)=f_0 + x\Delta f_0 + x(x-1)/2\ \Delta^2 f_0 + x(x-1)(x-2)/3!\ \Delta^3 f_0 \ ...$$

şeklinde verilir. Burada Δ ile verilen fark operatörüdür ve $\Delta f_i=f_{i+1}-f_i$ şeklindedir.

1.4.2 Newton Geriye Doğru Fark İnterpolasyon Formülleri

Bu defa $u=(x_n-x)/h$ ile belirlenen bir u değişkeni ve geriye doğru farklar operatörü kullanılarak interpolasyon polinomu

$$F(u)=f_n - \binom{u}{1}\nabla f_n + \binom{u}{2}\nabla^2 f_n + \ . \ . \ . \ + (-1)^n \binom{u}{n}\nabla^n f_n$$

şeklinde verilir. Burada ∇ geriye doğru farklar operatörü olup $\nabla f_i=f_i-f_{i-1}$ dir. İnterpolasyon formülünde kullanılacak olan farklar da tablo halinde yazılabilir.

i	x_i	$f(x_i)$	∇f_i	$\nabla^2 f_i$
0	x_0	$f(x_0)$		
			$\nabla f_1=f(x_1)-f(x_0)$	
1	x_1	$f(x_1)$		$\nabla^2 f_2= \nabla f_2-\nabla f_1$
			$\nabla f_2=f(x_2)-f(x_1)$	
2	x_2	$f(x_2)$		$\nabla^2 f_3= \nabla f_3-\nabla f_2$
			$\nabla f_3=f(x_3)-f(x_2)$	
3	x_3	$f(x_3)$		

h=1 için $u=x_n-x$ olur ve interpolasyon polinomu

$$F(x)=f_n - u\nabla f_n + u(u-1)/2!\nabla^2 f_n - \ldots+u(u-1)..(u-n+1)/n!\nabla^n f_n$$

olarak yazılabilir.

1.5 SPLİNE İNTERPOLASYON

Nümerik analizde klasik polinomla interpolasyon yaklaşımı (Lagrange İnterpolasyon gibi) artan örnek sayısı ile birlikte kullanılan polinomun derecesinin de artmasını gerektirir. Ayrıca bu yaklaşımların ciddi bazı sınırlamaları vardır. Yukarıda anlatılan interpolasyon metodları dışında interpolasyon fonksiyonu olarak sabit dereceli parçalı polinomlar kullanmak doğal bir gerekliliktir. Bu parçalı polinomların tamamı spline fonksiyon olarak adlandırılır.

2.1 B-SPLINE FONKSİYONLARI

Bu bölümde özellikle B-Spline fonksiyonları ile interpolasyon işlemi incelenecek ve fonksiyonların birbirlerine göre hata oranları karşılaştırılacaktır. Aşağıda yapılan analizde 0.dereceden 3. dereceye kadar B-spline fonksiyonlarının nasıl elde edildiği ve bu fonksiyonların temel özellikleri incelenmiştir.
Tek boyutlu ve sürekli bir $f(\xi)$ fonksiyonu için interpolasyon işlemi;

$$\hat{f}(\xi)=\sum_{k=1}^{K} c_k\, s_k(\xi) \text{ olarak yazılabilir.} \qquad (2.1)$$

c_k; giriş verilerinden elde edilen katsayıları, $s_k(\xi)$; temel fonksiyonu, K ise örnek (sample) sayısını göstermektedir.

Tanım: $\pi: \xi_0 < \xi_1 < \ldots < \xi_n < \xi_{n+1}$ reel eksen $[\xi_0, \xi_{n+1}]$ aralığındaki parçalar olsun. n. dereceden bir B-spline aşağıda yazıldığı gibi parçalı polinomlar şeklindedir.

$$B_n(\xi;\xi_0,\xi_1,\xi_2,\ldots\xi_{n+1})$$

$$=(n+1)\ (-1)^j \sum_{k=0}^{n+1} \frac{(\xi-\xi_k)^n\, u(\xi-\xi_k)}{\omega(\xi_k)} \qquad (2.2)$$

Bu eşitlikteki $$\omega(\xi_k)=\prod_{\substack{j=0\\ j\neq k}}^{n+1} (\xi_k-\xi_j) \qquad (2.3)$$

$$u(\xi-\xi_k)=\begin{cases}(\xi-\xi_k)^0 & \xi>\xi_k\\ 0 & \xi\le\xi_k\end{cases} \qquad (2.4)$$

$u(\xi-\xi_k)$ birim basamak fonksiyonudur.

n=0,1,2.... şeklinde değerler alabilir. Şekil 2.1 de ilk dört derecedeki B-spline'lar gösterilmektedir. Şekillerdeki $\Delta=(\xi_k-\xi_{k-1})$ dir. Yukarıda verilen 2.1 bağıntısından yararlanarak n=0,1,2,3 değerleri için parçalı polinomları yazabiliriz.

n=0 için;

$$B_0(\xi)=(1/\Delta)[(\xi-\xi_0)^0\,u(\xi-\xi_0)-(\xi-\xi_1)^0u(\xi-\xi_1)] \qquad (2.5)$$

n=1 için;

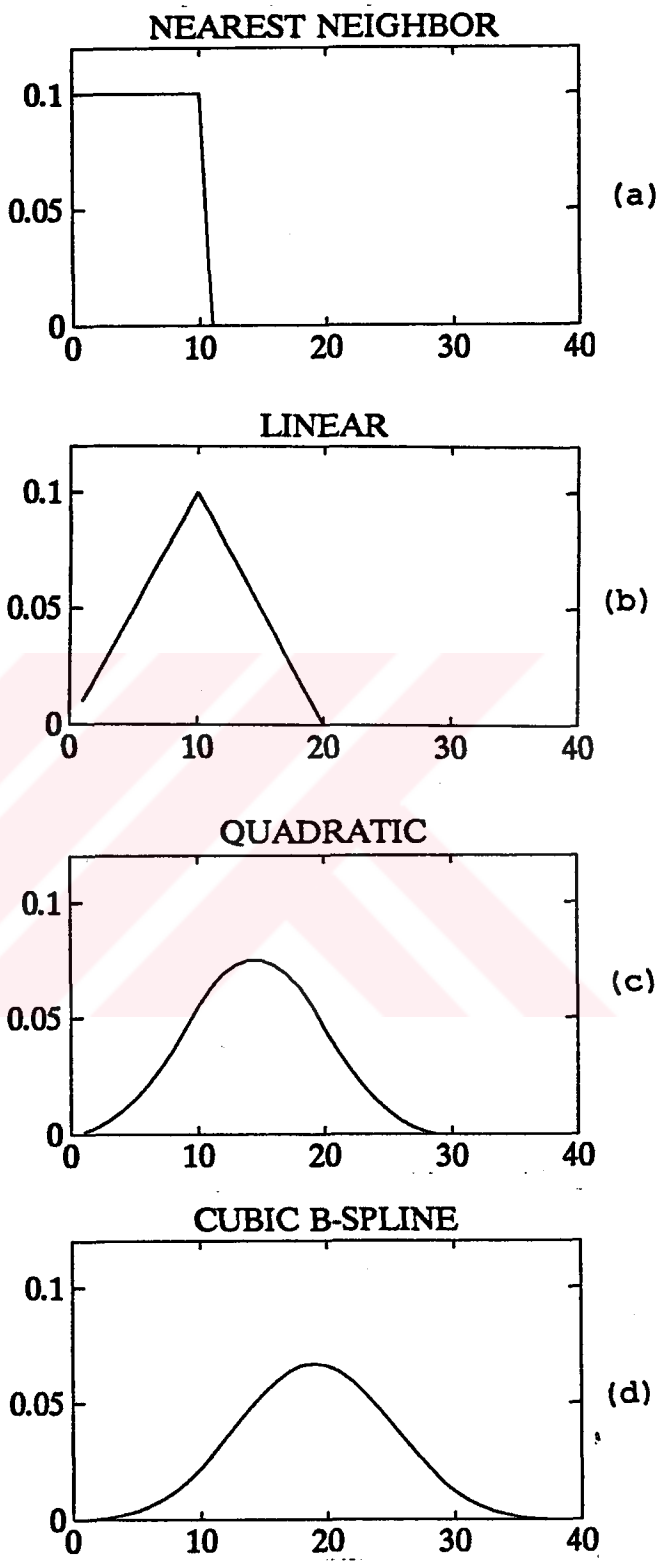
$$B_1(\xi)=(1/\Delta^2)[(\xi-\xi_0)\,u(\xi-\xi_0)-2(\xi-\xi_1)u(\xi-\xi_1)+(\xi-\xi_2)\,u(\xi-\xi_2)] \qquad (2.6)$$

n=2 için;

$$B_2(\xi)=(1/2\Delta^3)[(\xi-\xi_0)^2\,u(\xi-\xi_0)-3(\xi-\xi_1)^2u(\xi-\xi_1)+3(\xi-\xi_2)^2u(\xi-\xi_2)-(\xi-\xi_3)^2\,u(\xi-\xi_3)] \qquad (2.7)$$

n=3 için;

$$B_3(\xi)=(1/6\Delta^4)[(\xi-\xi_0)^3\,u(\xi-\xi_0)-4(\xi-\xi_1)^3u(\xi-\xi_1)+6(\xi-\xi_2)^3u(\xi-\xi_2)-4(\xi-\xi_3)^3u(\xi-\xi_3)+(\xi-\xi_4)^3\,u(\xi-\xi_4)] \qquad (2.8)$$



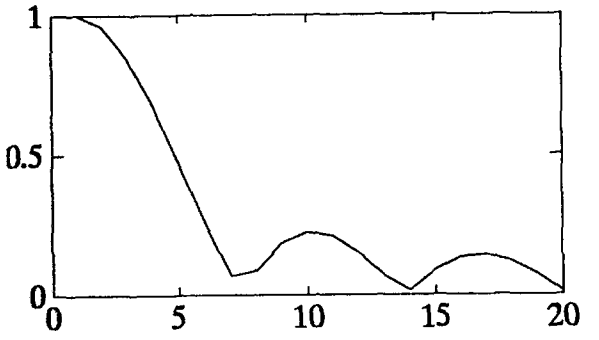
$B_0(\xi;\xi_k,\xi_{k+1})$
Örnekle-tut fonksiyonu

$B_1(\xi;\xi_{k-1},\xi_k,\xi_{k+1})$
Lineer B-spline

$B_2(\xi;\xi_{k-1},\xi_k,\xi_{k+1},\xi_{k+2})$
Quadratic B-spline

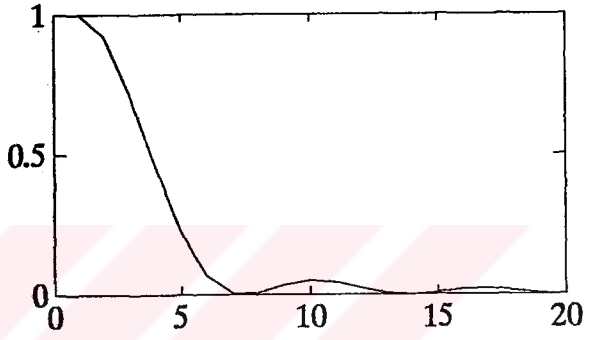
$B_3(\xi;\xi_{k-2},\xi_{k-1},\xi_k,\xi_{k+1},\xi_{k+2})$
Cubic B-spline

Şekil 2.1(a) B-Spline fonksiyonları



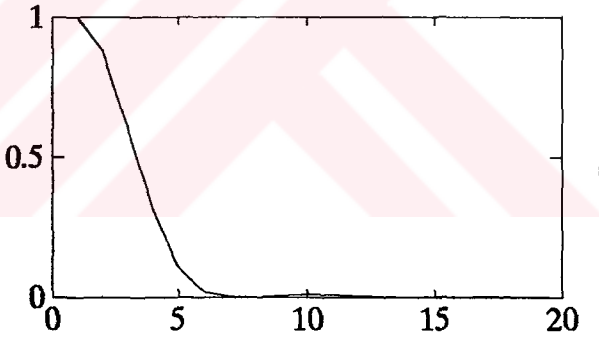
(a)

B_0
Örnekle-tut fonksiyonu



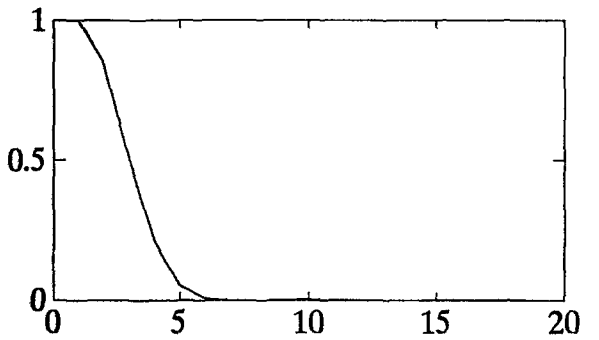
(b)

B_1
Lineer B-spline



(c)

B_2
Quadratic B-spline



(d)

B_3
Cubic B-spline

Şekil 2.1(b) B-Spline fonksiyonları fourier transformu

2.2 B-SPLİNE FONKSİYONLARININ ÖZELLİKLERİ

Düzgün aralıklarla verilen örnekler için denklem 2.2 den anlaşılacağı gibi B-spline fonksiyonları arasında konvolüsyon ilişkisi vardır.

$B_1=B_0*B_0$

$B_2=B_0*B_0*B_0$ (* konvolüsyon işlemini temsil eder.)

$B_3=B_0*B_0*B_0*B_0$

Şekil 2.1'dende görüldüğü gibi B_0-spline fonksiyonu bir örnekle tut fonksiyonudur ve örnek noktaları arasındaki süreksizliklerde kendisini tekrarlar.

B_1-spline fonksiyonu ile interpolasyon, örnek noktaları arasında iki parçalı ve lineer bir eğim gösterir.

B_2-spline fonksiyonu benzer şekilde grafik olarak birleştirilmiş parabollerden oluşur. Ardarda gelen parabollerin eğimleri birbiriyle süreklilik gösterir.

Şekil 2.1 de ki son fonksiyon olan B_3-spline fonksiyonu üçüncü dereceden parçalı polinomların birleşmesinden oluşur. Buradada ardarda gelen polinomların eğimleri birbirleriyle süreklilik gösterir.
Şekillerdende anlaşılacağı gibi B_0 ve B_1-spline fonksiyonları ile interpolasyon tatmin edici olmayacaktır. Diğer tarafta üçden daha büyük dereceli B-spline'larda daha fazla nokta birleştirilir ve işlem karmaşıklığı artar. Bu durumda interpolasyon işlemi polinomla interpolasyona benzer ve spline kullanmanın getirdiği avantaj ortadan kalkar. İyi bir interpolasyon sonucu ve

kolay hesaplanabilir olma özellikleri nedeniyle cubic B-splin'ı temel fonksiyon olarak seçmek iyi bir çözümdür. Cubic B-spline analitik olarak parçalı ve sadece beş nokta birleştirmesi nedeniyle çok esnektir. Ayrıca fonksiyon değerlerine çok iyi yaklaşıklılık sağlar.
Düzgün aralıklarla verilen örnek değerleri için cubic B-spline temel fonksiyonu aşağıdaki gibidir.

$$s(\xi-\xi_k) \equiv B_3(\xi;\xi_{k-2},\xi_{k-1},\xi_k,\xi_{k+1},\xi_{k+2})$$
$$= (1/6\Delta^4)\,[(\xi-\xi_{k-2})^3\,u(\xi-\xi_{k-2}) - 4(\xi-\xi_{k-1})^3 u(\xi-\xi_{k-1})$$
$$+6(\xi-\xi_k)^3 u(\xi-\xi_k) - 4(\xi-\xi_{k+1})^3 u(\xi-\xi_{k+1}) + (\xi-\xi_{k+2})^3\,u(\xi-\xi_{k+2})] \qquad (2.9)$$

B-spline temel fonksiyonları Şekil 2.1 'dende görüldüğü gibi tamamiyle pozitiftir. B-spline fonksiyonlarının bu özelliği görüntü işleme problemlerinde, görüntüyü oluşturan noktaların ışık şiddeti veya yansımaları herzaman pozitif olduğu için çok kullanışlıdır.

2.3 B-SPLİNE İNTERPOLASYON TEORİSİ

Bu bölümde Quadratic ve Cubic B-spline temel fonksiyonları için interpolasyon formülleri elde edecek ve tek boyutlu işaretler için interpolasyon sonuçları karşılaştırılacaktır.

2.3.1 Quadratic B-Spline İnterpolasyon Teorisi

$f(\xi)$ için $\xi=\xi_k+\Delta X$ olduğunu düşünerek $0\le X\le 1$ için denklem 2.7 yi kullanarak ve Şekil 2.2 den faydalanarak $f(\xi)$ (2.1) formülü;

$$s(\xi-\xi_k) \equiv B_2(\xi;\xi_{k-1},\xi_k,\xi_{k+1},\xi_{k+2})$$
$$= 1/2\Delta^3\,[(\xi-\xi_{k-1})^2\,u(\xi-\xi_{k-1}) - 3(\xi-\xi_k)^2 u(\xi-\xi_k) + 3(\xi-\xi_{k+1})^2 u(\xi-\xi_{k+1})$$
$$-(\xi-\xi_{k+2})^2\,u(\xi-\xi_{k+2})] \qquad (2.10)$$

$$\hat{f}(\xi)=1/2\Delta^3\{c_{k-1}[(\xi-\xi_{k-2})^2-3(\xi-\xi_{k-1})^2+3(\xi-\xi_k)^2]+$$
$$c_{k+0}[(\xi-\xi_{k-1})^2-3(\xi-\xi_k)^2]+$$
$$c_{k+1}[(\xi-\xi_k)^2]\} \qquad (2.11)$$

şeklinde yazılabilir. $\xi=\xi_k+\Delta X$ özelliği denklemde kullanılırsa;

$$\hat{f}(\xi_k+\Delta X)=1/2\Delta\{c_{k-1}[(2+X)^2-3(1+X)^2+3X^2]+$$
$$c_k[(1+X)^2-3X^2]+c_{k+1}X^2\} \qquad (2.12)$$

Burada çift dereceli B-spline larda 1/2 öteleme gerekliliği ortaya çıkar. Denklem daha basit olarak yazılırsa;

$$\hat{f}(\xi_k+\Delta X)=1/8\Delta\sum_{i=0}^{2} b_i x^{2-i}$$

olur. Bu eşitlikte $b=\Omega c_k$ $b=[b_0,b_1,b_2]^t$ ve $c=[c_{k-1},c_k,c_{k+1}]^t$

$$\Omega=\begin{bmatrix}4 & -8 & 4\\ -4 & 0 & 4\\ 1 & 6 & 1\end{bmatrix}$$

dır. Örnek değeri için X=0 alınır ve $\xi=\xi_k$ için interpolasyon yapılırsa;

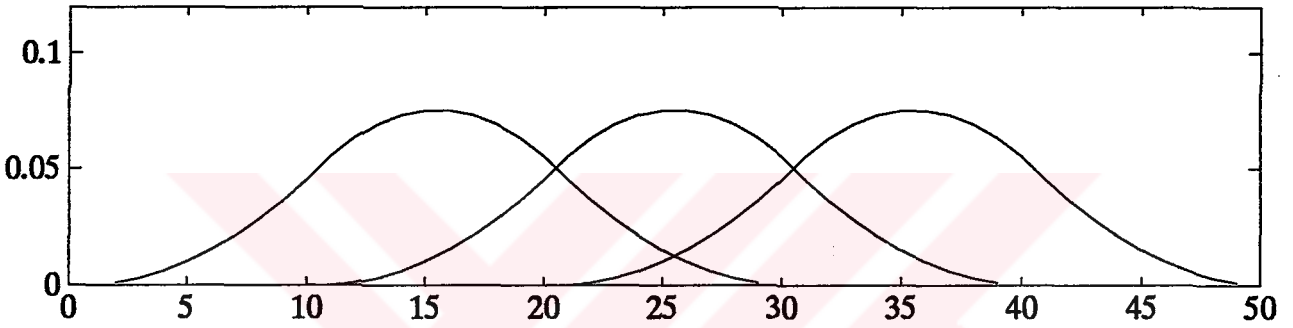
$$\hat{f}(\xi_k)=1/2\Delta(c_{k-1}+c_k)$$

bulunur.Bu durumda interpolasyon ifadesi vektör şeklinde yazılabilir.

$$\hat{f}=EC$$

$$E=1/8\Delta \begin{bmatrix} 6 & 1 & 0 & \cdots & & \\ 1 & 6 & 1 & \cdots & & \\ 0 & 1 & 6 & 1 & \cdots & \\ 0 & 0 & 1 & 6 & 1 & \cdots \\ 0 & 0 & 0 & 1 & 6 & 1 \\ 0 & 0 & 0 & 0 & 1 & 6 \end{bmatrix}$$

E matrisi üçlü köşegen kare bir matristir. c katsayılar matrisi örnek noktaları için oluşturulur ($c=E^{-1}f$) ve interpolasyon noktaları için oluşturulan bu katsayılar matrisi kullanılır.



Şekil 2.2 $f(\xi)$ fonksiyonu için Quadratic B-spline İnterpolasyon

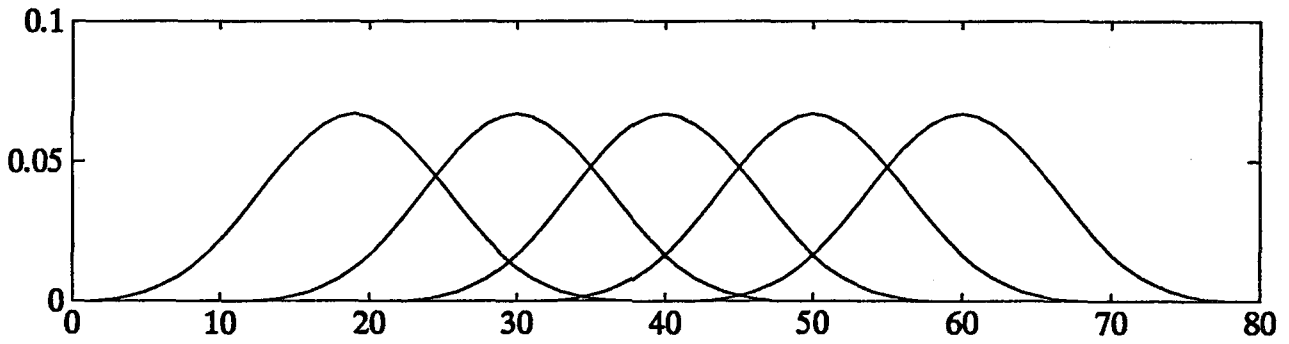
2.3.2 Cubic B-Spline interpolasyon Teorisi

$\hat{f}(\xi)$ için $\xi=\xi_k+\Delta X$ olduğunu düşünerek $0\leq X\leq 1$ için denklem 2.9 u kullanarak ve Şekil 2.3 den faydalanarak $\hat{f}(\xi)$ (2.1) formülü yazılırsa;

$$\begin{aligned}\hat{f}(\xi)=(1/6\Delta^4)\{&c_{k-1}[(\xi-\xi_{k-3})^3-4(\xi-\xi_{k-2})^3+6(\xi-\xi_{k-1})^3-4(\xi-\xi_k)^3]+\\ &c_{k+0}[(\xi-\xi_{k-2})^3-4(\xi-\xi_{k-1})^3+6(\xi-\xi_k)^3]+\\ &c_{k+1}[(\xi-\xi_{k-1})^3-4(\xi-\xi_k)^3]+\\ &c_{k+2}(\xi-\xi_k)^3\ \}\end{aligned} \qquad (2.13)$$

şeklindedir. $\xi=\xi_k+\Delta X$ özelliği denklemde kullanılırsa;

$$\hat{f}(\xi_k+\Delta X)=1/6\Delta\{c_{k-1}[(3+X)^3-4(2+X)^3+6(1+X)^3-4X^3]+$$
$$c_k[(2+X)^3-4(1+X)^3+6X^3]+$$
$$c_{k+1}[(1+X)^3-4X^3]+c_{k+2}X^3\} \qquad (2.14)$$



Şekil 2.3 f(ξ) fonksiyonu için Cubic B-spline İnterpolasyon

Bu denklem daha basit olarak yazılırsa;

$$\hat{f}(\xi_k+\Delta X)=1/6\Delta\sum_{i=0}^{3} b_i x^{3-i} \quad 0\le X\le 1$$

olur. Bu eşitlikte $b=\Omega c_k$ $b=[b_0,b_1,b_2,b_3]^t$ ve $c=[c_{k-1},c_k,c_{k+1},c_{k+2}]^t$ dir.

$$\Omega=\begin{bmatrix}1&3&3&1\\3&6&3&0\\3&0&3&0\\1&4&1&0\end{bmatrix}$$

Örnek değeri için X=0 alınır ve $\xi=\xi_k$ için interpolasyon yapılırsa;

$$\hat{f}(\xi_k)=1/6\Delta(c_{k-1}+4c_k+c_{k+1})$$

bulunur. Bu durumda interpolasyon ifadesi vektör şeklinde yazılırsa;

$$\hat{f}=EC \quad \text{olur.}$$

$$E=1/6\Delta \begin{bmatrix} 4 & 1 & 0 & \cdots & \\ 1 & 4 & 1 & 0 & \cdots \\ 0 & 1 & 4 & 1 & \cdots \\ 0 & 0 & 1 & 4 & 1 \cdots \\ 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

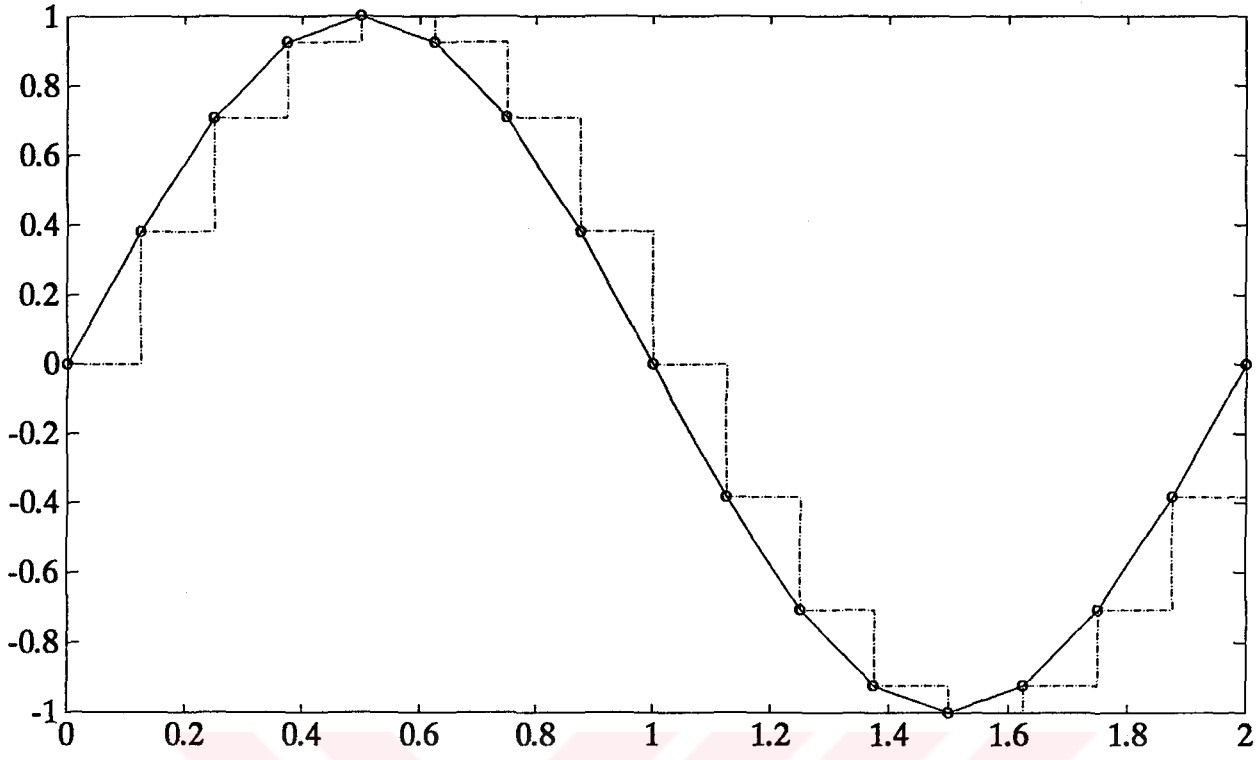
E matrisi üçlü köşegen kare bir matristir. c katsayılar matrisini oluşturmak için örnek noktaları vektörü(f) ile E matrisinin tersi çarpılır ($c=E^{-1}f$) .E matrisi üçlü köşegen bir matris olduğu için tersinin alınması için basit bazı yöntemler vardır. interpolasyon noktaları için oluşturulan bu katsayılar matrisi kullanılır.

Aşağıdaki şekillerde sırasıyla 0.dereceden 3.dereceye kadar B-spline lar için 17 örnek değeri için interpolasyon sonuçları gösterilmektedir.

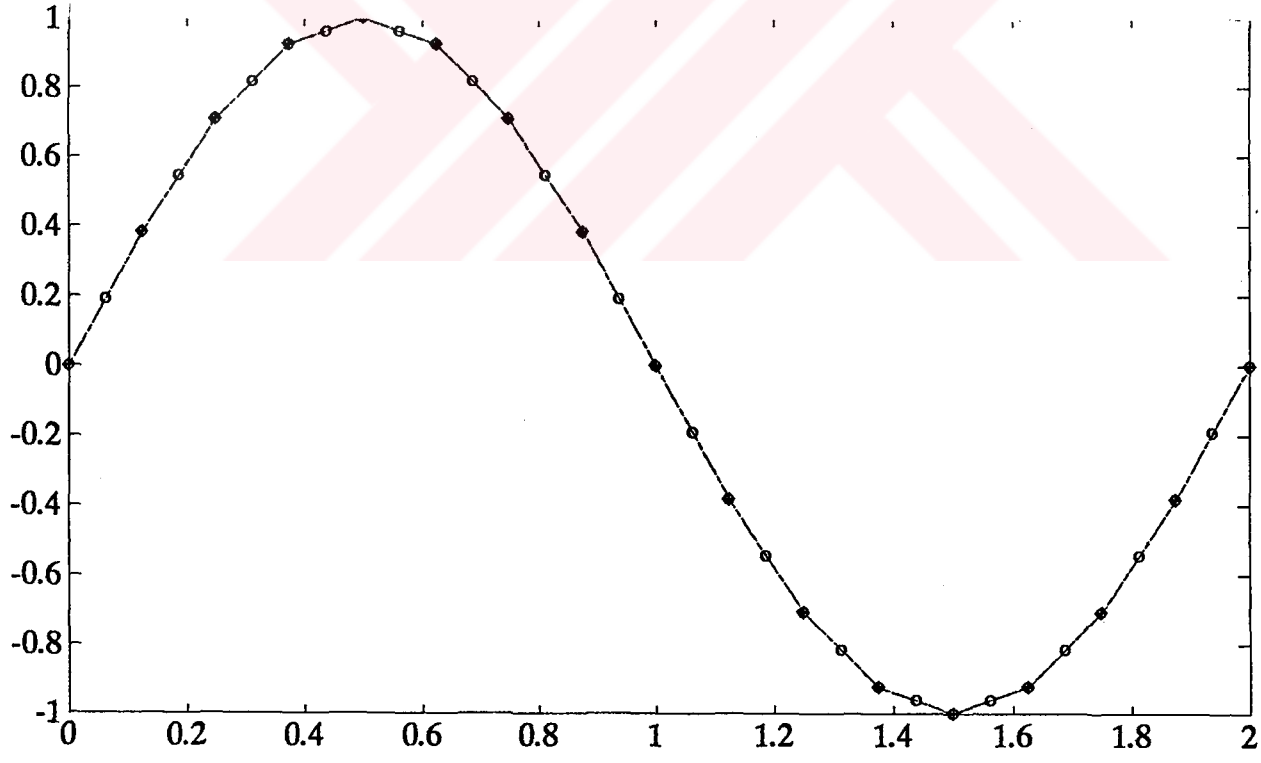
Şekil 2.4.(a) da örnekle tut fonksiyonu interpolasyon sonucu kesikli çizgilerle gösterilmiştir.

Şekil 2.4.(b) de lineer B-spline interpolasyon için interpolasyon noktaları örnek noktalarının tam ortasında seçilmiştir.

Aynı interpolasyon noktaları ile ilgili hata fonksiyonları Şekil 2.5 de sırasıyla birinci dereceden üçüncü dereceye kadar B-spline lar için gösterilmektedir.



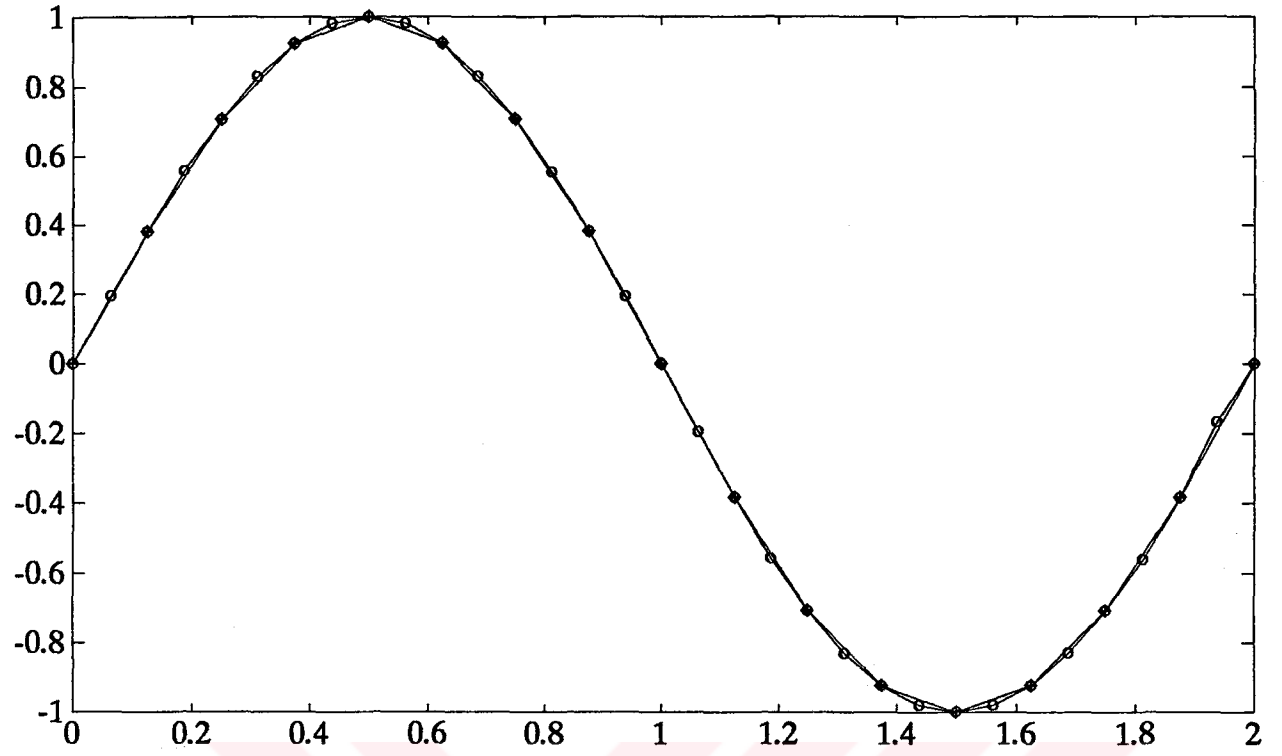
Şekil 2.4 (a) 0.dereceden B-spline (Örnekle-Tut)



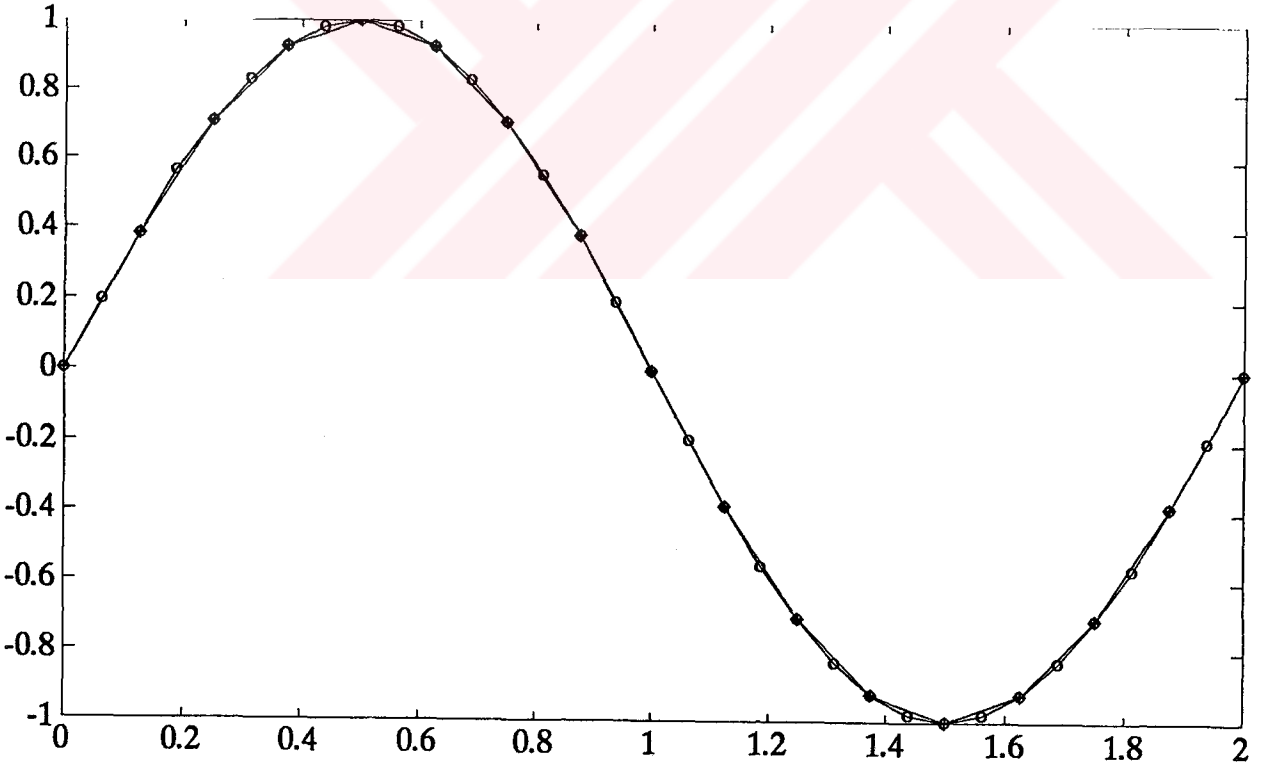
Şekil 2.4(b) Lineer interpolasyon

o Örnek değerlerini

⊕ İnterpolasyon değerlerini gösterir.



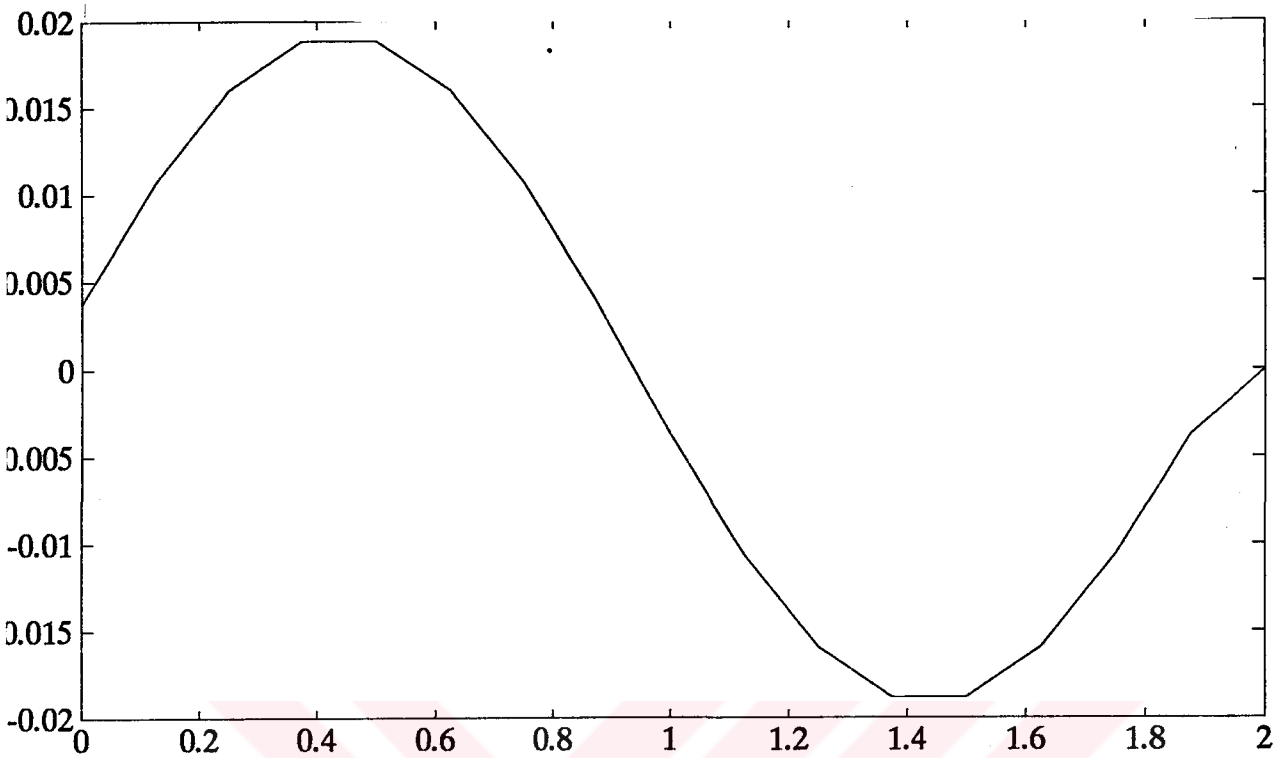
Şekil 2.4(c) Quadratic B-spline interpolasyon



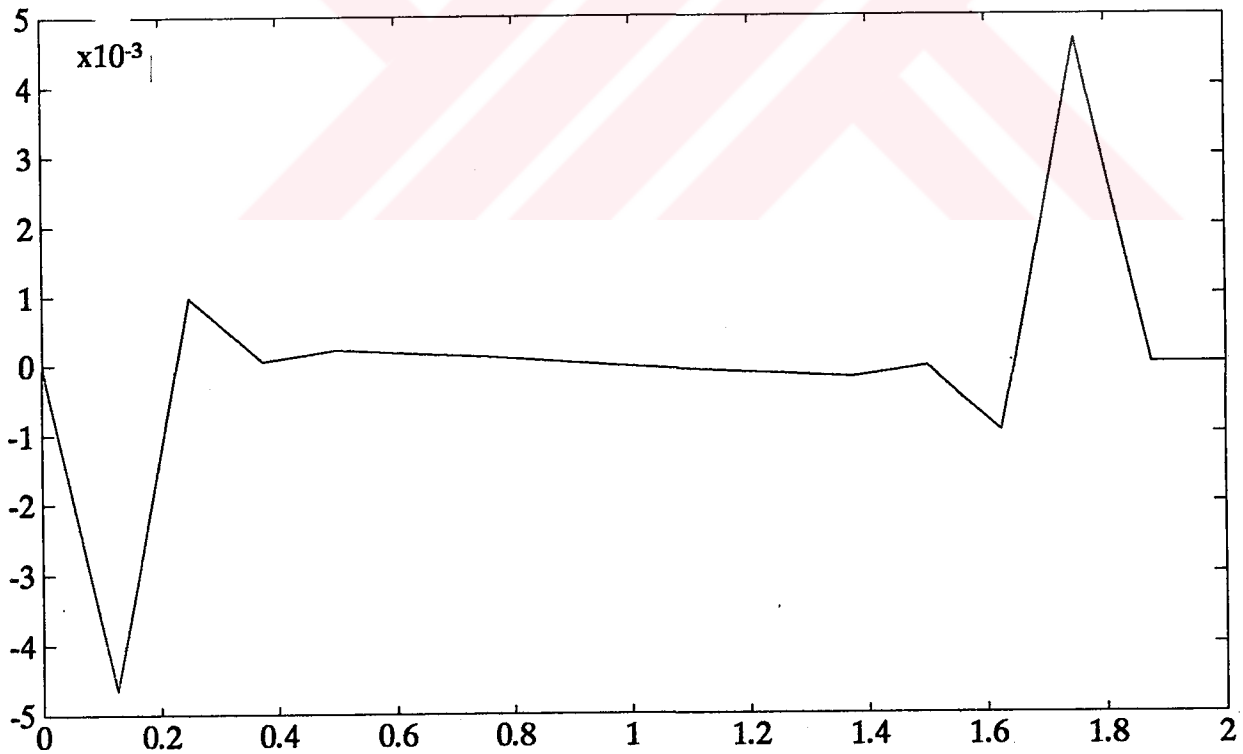
Şekil 2.4(d) Cubic B-spline interpolasyon

o Örnek değerlerini

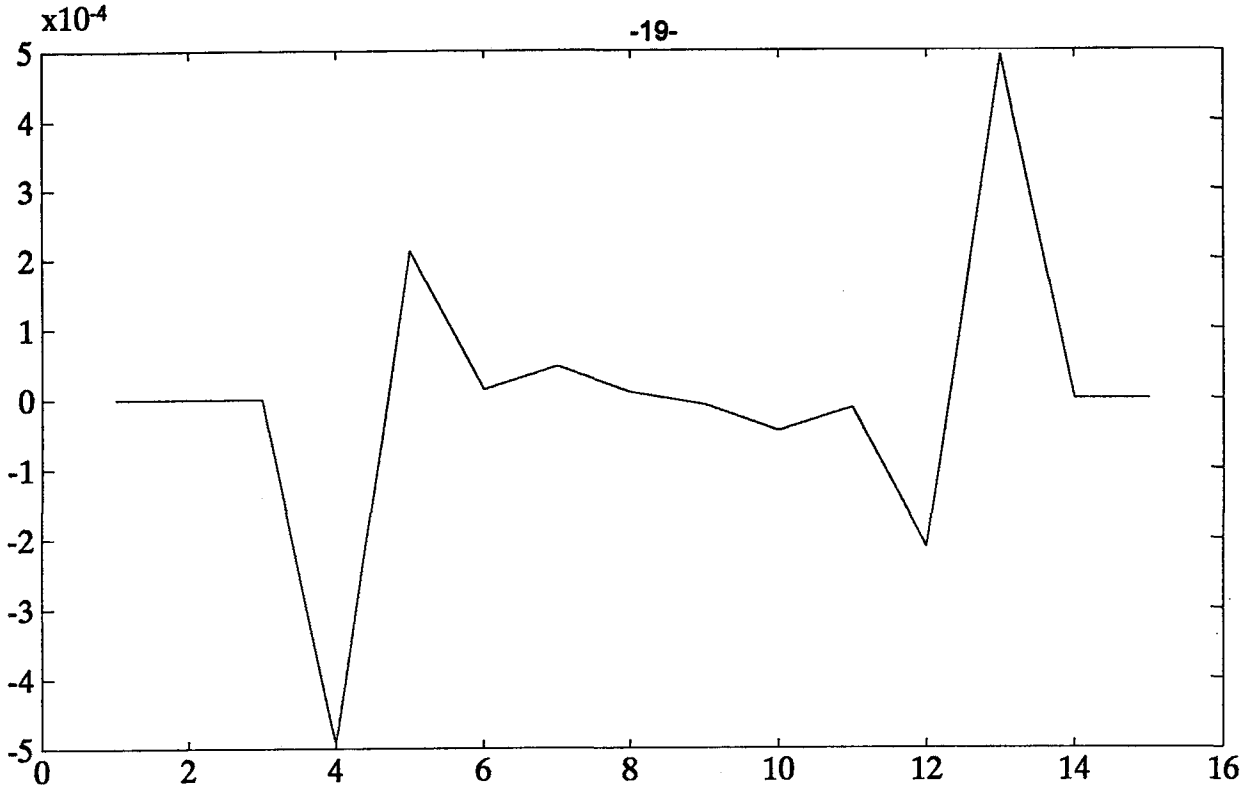
⊕ İnterpolasyon değerlerini gösterir.



Şekil 2.5 (a) Lineer interpolasyon hata fonksiyonu



Şekil 2.5 (b) Quadratic B-spline İnterpolasyon hata fonksiyonu



Şekil 2.5(c) Cubic B-spline interpolasyon hata fonksiyonu

2.4 İKİ BOYUTLU İŞARETLERDE CUBİC B-SPLİNE İNTERPOLASYON

Görüntü üzerinde interpolasyon, pek çok işaret işleme uygulamalarında önemli bir rol oynar. Bu işaret işleme uygulamaları, rezolüsyon değiştirme, görüntü boyutlarını değiştirme gibi konularda olabilir. Modern display workstation'larında bazı detayları daha iyi görebilmek veya daha iyi bir genel görünüş için görüntü boyutlarını değiştirmek amacı ile verimli bir ölçekleme düzenine ihtiyaç duyulur.

Denklem 2.1 de tek boyutlu işaretler için verilen interpolasyon fonksiyonu iki boyutlu işaretler için geliştirilirse;

$$\hat{f}(\xi,\eta) = \sum_{k=1}^{K} \sum_{l=1}^{L} c_k \, s_k(\xi) \, s_l(\eta) \quad \text{olarak yazılabilir.} \qquad (2.15)$$

Burada $f(\xi,\eta)$ için $\xi=\xi_k+x\Delta$ ve $\eta=\eta_l+y\Delta$ olduğunu düşünerek $0\le x\le 1$ ve $0\le y\le 1$ için;

$$\hat{f}(x,y)=1/(6*\Delta)\{\hat{f}_{l-1}(x)[(3+y)^3-4(2+y)^3+6(1+y)^3-4y^3] + \hat{f}_l(x)[(2+y)^3-4(1+y)^3+6y^3] + \hat{f}_{l+1}(x)[(1+y)^3-4y^3]\}$$

yazılabilir. Burada

$$\hat{f}_j(x)= 1/(6*\Delta)\sum_{i=0}^{3} b_{ij}x^{3-i} \quad j=l-1\ ,\ l\ ,\ l+1\ ,\ l+2$$

$$b_j=\Omega c_j \quad b_j=(b_{0j},b_{1j},b_{2j},b_{3j})^t$$
$$c_j=(c_{k-1,j},c_{k,j},c_{k+1,j},c_{k+2,j})^t$$

dir. Ω daha önce tek boyutlu analizde verilenle aynıdır. c matrisini oluşturabilmek için örnek noktası olan (ξ_k,η_l) noktası için interpolasyon yaparsak

$$\hat{f}(\xi_k,\eta_l)=1/(36*\Delta^2)[(c_{k-1,l-1}+4c_{k,l-1}+c_{k+1,l-1})+ 4(c_{k-1,l}+4c_{k,l}+c_{k+1,l})+ (c_{k-1,l+1}+4c_{k,l+1}+c_{k+1,l+1})]$$

şeklinde yazılabilir. Bu ifade bütün $k=1,2,...K$ ve $l=1,2,...L$ için geçerlidir. Bu eşitliği matris formunda yazarsak;

$$F=ECE$$

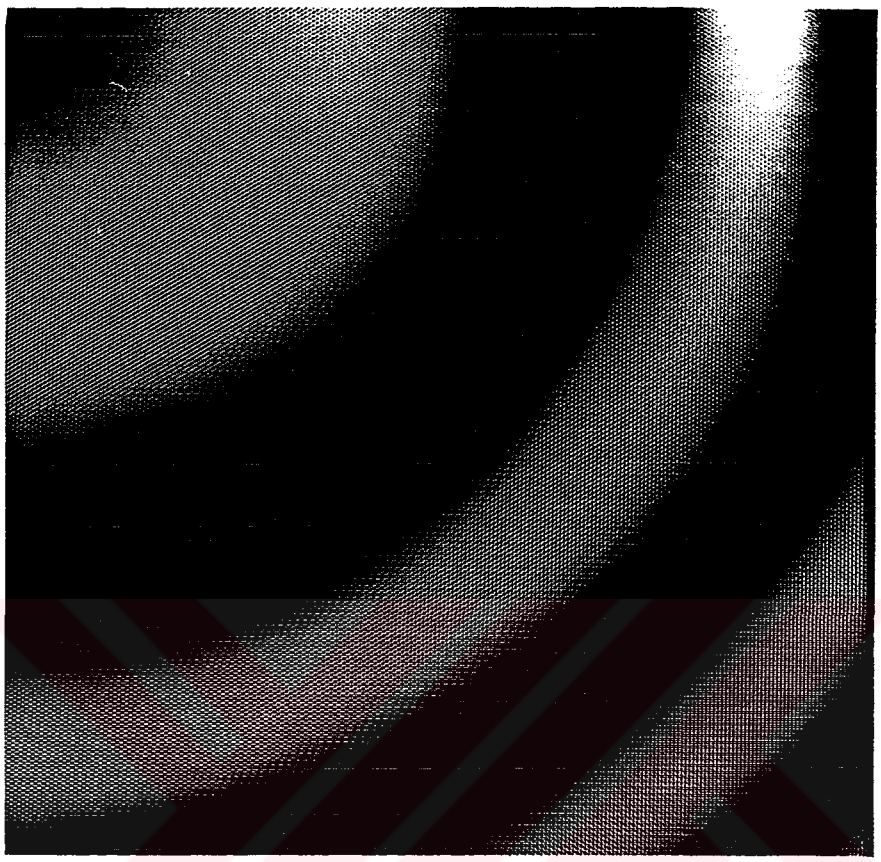
olur. Verilen örnek değerleri için C katsayılar matrisi hesaplanabilir (C=E'FE').

İki boyutlu işaretler için geliştirilen bu yöntem sayısal bir örnekle de gösterilebilir.

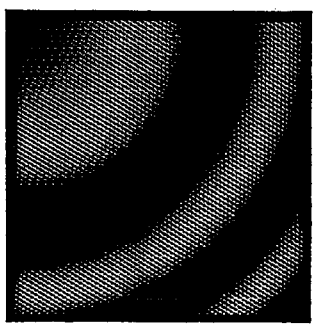
Merkezden çevreye doğru simetrik $f(x,y)=\sin(0.5r^2)$ gibi bir fonksiyon için inceleme yapalım. Burada $r^2=x^2+y^2$ dir. Bu yapay görüntü, fonksiyonun verdiği genlik değerlerinin ışık şiddetine çevrilmesi ile oluşturulur. Maksimum ışık şiddeti olan beyaz, fonksiyonun maksimum değeri olan +1'e karşı gelir. Sıfır ışık şiddeti olan siyah, fonksiyonun minimum değeri olan -1'e karşı gelir. Fonksiyonun ara değerleri ise gri seviyeler olarak ifade edilir.

Şekil 2.6(a)da 150x150 pixel büyüklüğünde verilen orjinal görüntü eşit aralıklarla örneklenmiş ve Şekil 2.6(b)'de görülen 50x50 pixel büyüklüğündeki şekil elde edilmiştir. Cubic B-spline interpolasyon işleminde küçültülen resim kullanılmış ve sonuç Şekil 2.6 (c) de gösterilmiştir. Şekil 2.6(d)'de görülen cubic B-Spline interpolasyon hatasıdır. Görüntü üzerinde uygulanan bu interpolasyon işlemi sayısal görüntü büyütme işlemi olarak değerlendirilebilir. Şekil 2.7'de 0.dereceden B-spline interpolasyon sonucu ve hatası gösterilmiştir. Gösterilen hata, orjinal şekil ile interpolasyon işleminden elde edilen şekil arasındaki farkın mutlak değeridir. Bu şekillerde siyah hatanın sıfır olduğunu, beyaz mutlak hatanın maksimum olduğunu gösterir. Şekil 2.6(d) Cubic B-spline'la interpolasyon hatasını,Şekil 2.7(b) 0.dereceden B-spline'la interpolasyon hatasını gösterir.

Şekil 2.8(a)'da ise scanner dan alınan 100x100 pixel büyüklüğündeki arapça bir metin gösterilmektedir. 2.8(b)'deki 50x50 pixel büyüklüğündeki şekil orjinal resmin eşit aralıklarla örneklenmesinden elde edilmiştir.Cubic B-spline interpolasyon işleminde küçültülen resim kullanılmış ve sonuç Şekil 2.8(c) de gösterilmiştir. Şekil 2.8(d)'de görülen 0.dereceden B-Spline interpolasyon sonucudur. Şekil 2.9'da Şekil 2.8'deki şekiller için interpolasyon hatası gösterilmiştir.

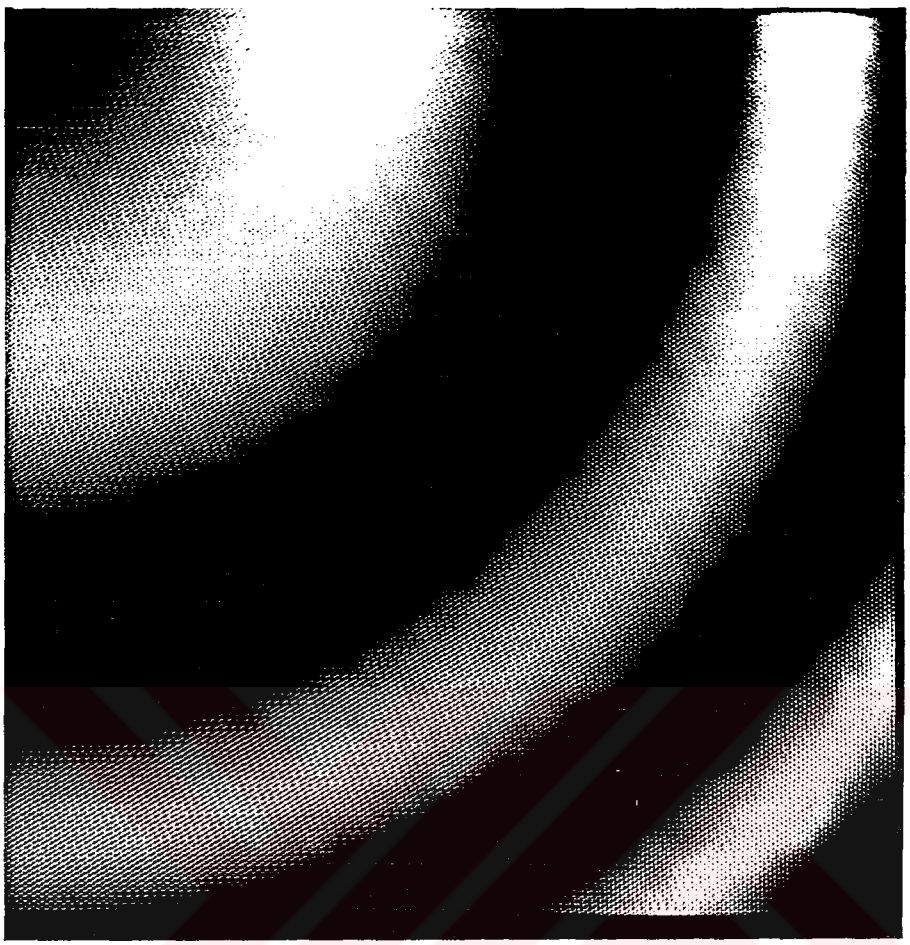


(a)

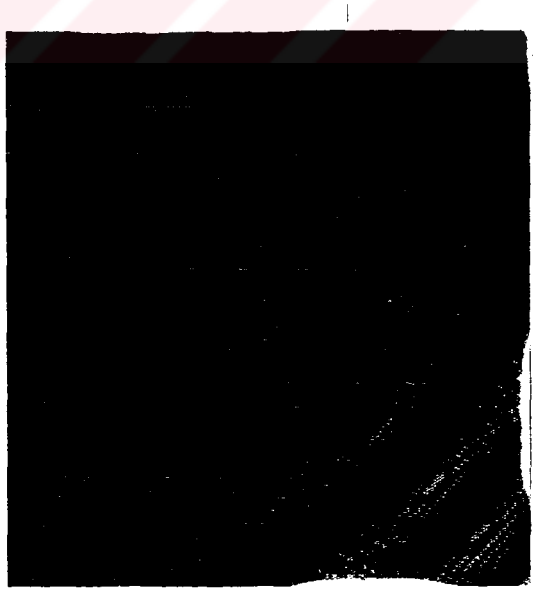


(b)

Şekil 2.6(a)(b) $f(x,y)=\sin(0.5r^2)$ fonksiyonu

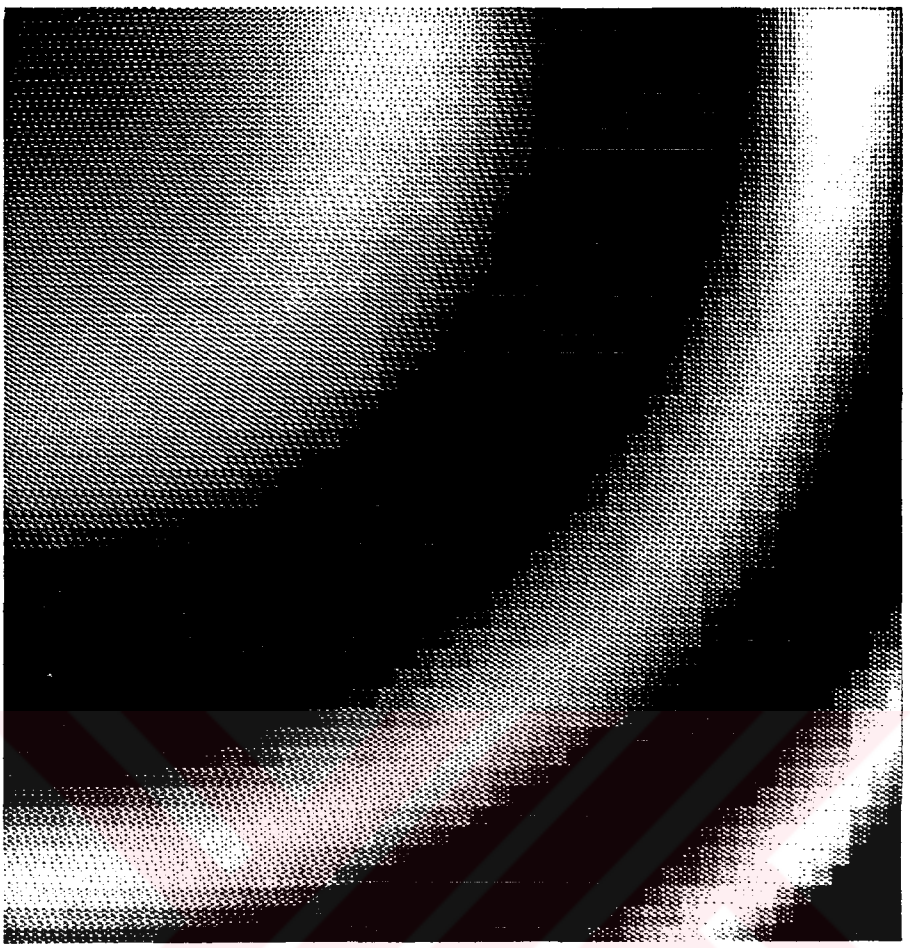


(c)

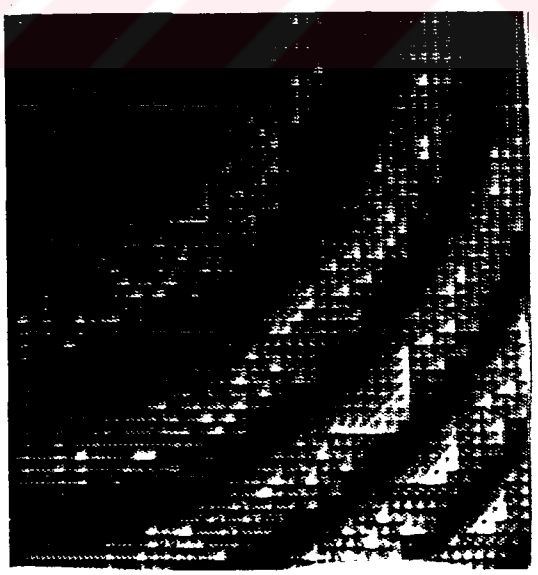


(d)

Şekil 2.6(c) $f(x,y)=\sin(0.5r^2)$ fonksiyonu ile ilgili Cubic B-Spline interpolasyon sonucu ve (d) hata

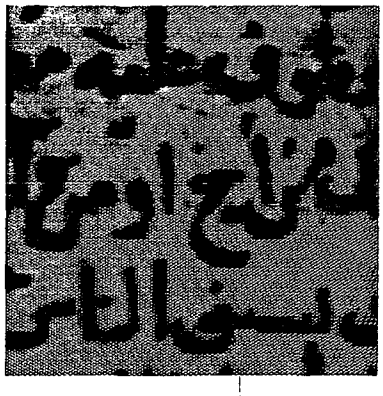


(a)

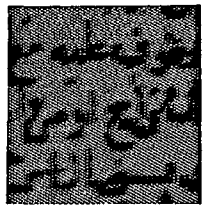


(b)

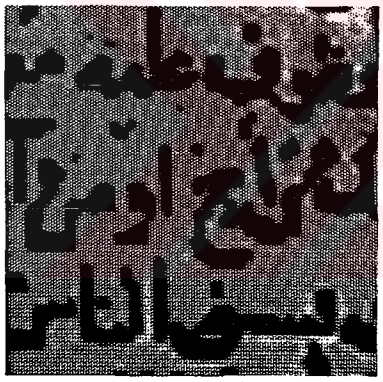
Şekil 2.7(a) $f(x,y)=\sin(0.5r^2)$ fonksiyonu ile ilgili
0.derece B-spline interpolasyon sonucu ve (b) hata



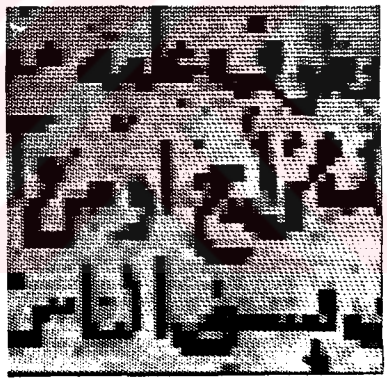
(a)



(b)

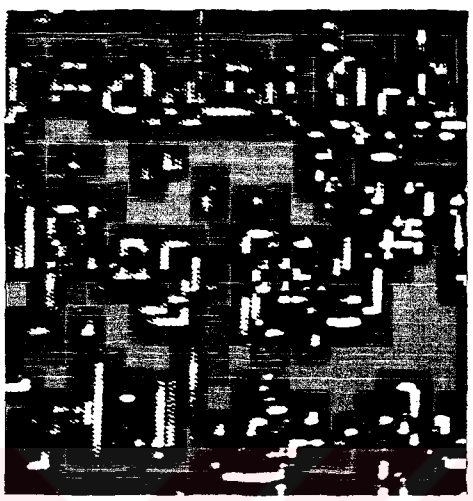


(c)

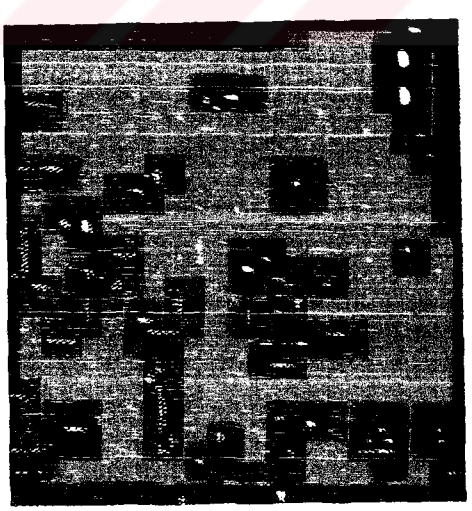


(d)

Şekil 2.8 (a) Scanner'dan alınan 100x100 pixel arapça metin
(b) a'daki resmin küçültülmüş şekli
(c) Cubic B-spline interpolasyon sonucu
(d) 0.dereceden B-spline interpolasyon sonucu

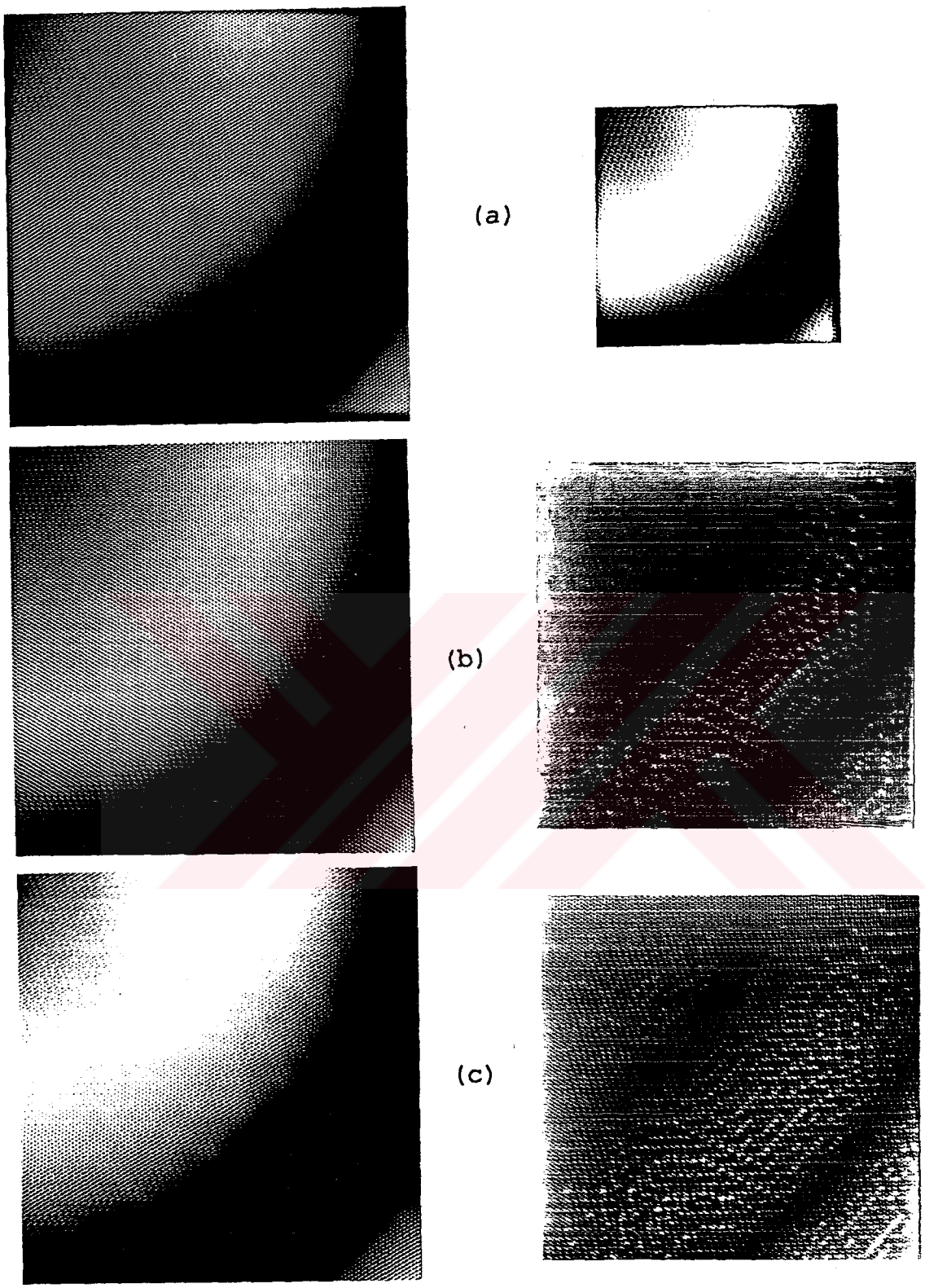


(a)



(b)

Şekil 2.9(a) Cubic B-spline'la interpolasyon hatası
(b) 0.dereceden B-spline'la interpolasyon hatası



Şekil 2.10 (a) Orjinal Şekil
(b) 9x9 Pencere Uygulaması ve hata oranı
(c) 7x7 Pencere Uygulaması ve hata oranı

3.1 B-SPLINE FİLTRE İLE İNTERPOLASYON

Bütün interpolasyon işlemlerinde ortak olan özellik, ayrık noktalar topluluğundan bir sürekli işareti ifade edecek parametrelerin oluşturulmasıdır. En basit yaklaşımlar 0.dereceden (nearest neighbor) ve 1.dereceden (bilineer) interpolasyonlardır. Cubic Spline interpolasyon metodu Hou ve Andrews tarafından tanıtılmış, çok gelişmiş bir tekniktir. Bu tekniğin karmaşık bir hesap gerektirmesi nedeniyle, araştırmacılar cubik konvolüsyon gibi bazı alternatif teknikler önermişlerdir.

B-spline interpolasyon metodları iki temel nedenden dolayı fazla kullanılmamaktadır. Öncelikle işaret işleme ve görüntü işlemede anlatılan spline katsayılarının belirlenmesi için kullanılan algoritma yeterince verimli değildir. İkincisi B-spline'ların görüntü çözünülürlüğünü (rezolüsyon) düşürdüğüne inanılır. Fakat bu yanlış yorumlar aslında yanlış uygulamaların bir sonucudur. Örneğin bazı uygulamalarda başlangıç örnek değerleri B-spline katsayıları yerine yazılarak interpolasyon işlemi yapılır ki buda hatanın büyük olmasına neden olur.

Bu bölümde tanıtılacak olan Doğrudan Spline Dönüşümü (genişleme katsayılarının belirlenmesi işlemi) ve dolaylı Spline dönüşümlerinin (seçimlik bir interpolasyonla orjinal örneklenmiş değerleri yeniden oluşturma işlemi) en önemli avantajı basit bir filtre operasyonu olarak yorumlanabilmeleridir. Bu filtrelerin rekürsif yapısından faydalanarak hızlı hesaplanabilen algoritmalar türetilebilir. Herhangi bir dereceden bir B-spline interpolatörü frekans ve impuls cevabı cinsinden karakterize etmek için basit prosedürler tanımlanabilir.

3.2 AYRIK (DISCRET) B-SPLINE

Ayrık B-spline'lar, sürekli B-spline fonksiyonlarının örneklenmesi ile oluşturulur. Yatay genişletme faktörü m olan n.dereceden ayrık B-spline aşağıdaki gibi yazılabilir.

$$b_m^n(k) = 1/m^n \sum_{j=0}^{n+1} (-1)^j /n! \binom{n+1}{j} (k-jm)^n \mu(k-jm) \qquad (3.1)$$

Bu fonksiyonları z-dönüşümleri şeklinde göstermek için önce kuvvet serisi dönüşümlerine ihtiyaç duyulur.
$\{k^n;\ k=0,1,....+\infty\}$ şöyle tanımlanır;

$$P^n(z) = \sum_{k=0}^{+\infty} k^k z^{-k} = \sum_{k=-\infty}^{+\infty} k^n z^{-k} \mu(k) \qquad (3.2)$$

$\mu(k)$ nın z dönüşümünü kullanarak;

$$P^0(z) = 1/(1-z^{-1})$$

yazılabilir.Buna göre bütün diğer kuvvet serilerinin transformunu geliştirebiliriz. Genel terim;

$$P^n(z) = A^n(z)/(1-z^{-1})^{n+1}$$

olarak kolaylaştırılabilir. Buna göre denklem 3.1 in z dönüşümü $P^n(z)$ ifadesininde kullanılmasıyla şöyle olur;

$$B_m^n(z) = 1/m^n \sum_{j=0}^{n+1} 1/n! \binom{n+1}{j} (-1)^j z^{-jm} A^n(z)/(1-z^{-1})^{n+1} \qquad (3.3)$$

$(-1)^j z^{-jm} = (-z^{-m})^j$ olduğundan ve $(x+1)^{n+1} = \sum_{j=0}^{n+1} \binom{n+1}{j} x^j$

olduğundan

$$B_m^n(z) = 1/m^n \; A^n(z)/n!\,[(1-z^{-m})/(1-z^{-1})]^{n+1} \qquad (3.4)$$

yazılabilir. Bu ifade daha kullanışlı bir şekilde yazılabilir.

$$B_m^n(z) = 1/m^n \; B_1^n(z)\,(B_m^n(z))^{n+1} \qquad (3.5)$$

$$B_1^n(z) = A^n(z)/n! = \sum_{k=0}^{n+1} b_1^n(k)\, z^{-k} \qquad (3.6)$$

$$B_m^0(z) = (1-z^{-m})/(1-z^{-1}) = \sum_{k=0}^{m-1} z^{-k} \qquad (3.7)$$

Aşağıda n=1 den 3'e kadar hesaplanan B-spline z transformları tablo 3.1'de n=5'e kadar gösterilmiştir. Bu incelemede genişleme faktörü m=1 alınmıştır.

n=1 için

$$b_1^1(k) = \sum_{j=0}^{2} (-1)^j/1! \binom{2}{j} (k-j)\,\mu(k-j)$$

$$b_1^1(k) = k\mu(k) - 2(k-1)\mu(k-1) + (k-2)\mu(k-2)$$

$$B_1^n(z) = z^{-1} = A^1(z)$$

$$P^1(z)=A^1(z)/(1-z^{-1})^2= z^{-1}/(1-z^{-1})^2$$

n=2 için

$$b_1^2(k)= \sum_{j=0}^{3} (-1)^j/2! \binom{3}{j} (k-j)^2 \mu(k-j)$$

$$b_1^2(k)=k^2\mu(k)-3(k-1)^2\mu(k-1)+3(k-2)^2\mu(k-2)-(k-3)^2\mu(k-3)$$

$$B_1^2(z) =1/2(z^{-1}+z^{-2})= A^2(z)/2!$$

$$P^2(z)=A^2(z)/(1-z^{-1})^2= (z^{-1}+z^{-2})/(1-z^{-1})^3$$

n=3 için

$$b_1^3(k)= \sum_{j=0}^{4} (-1)^j/3! \binom{4}{j} (k-j)^3 \mu(k-j)$$

$$b_1^3(k)=1/6[k^3\mu(k)-4(k-1)^3\mu(k-1)+6(k-2)^3\mu(k-2)-4(k-3)^2 \mu(k-3)+(k-4)^3\mu(k-4)$$

$$B_1^3(z) =1/6(z^{-1}+4z^{-2}+z^{-3})= A^2(z)/3!$$

Cubic B-spline için doğrudan B-spline transformunu (ters filtre) yazarsak;

$$S_n(z)=B_1^n(z)^{-1}=6/(z^{-1}+4z^{-2}+z^{-3})$$

olur. Bu denklemin kökleri ise; $z_1=\alpha=-2+\sqrt{3}$ ve $z_2=1/\alpha=-2-\sqrt{3}$ şeklindedir.

Tablo 3.1 n=0 dan 5'e kadar Ayrık B-spline ların Z transformları

n	$P^n(z)$	$A^n(z)$	$B^n{}_1(z)$
0	$A^0(z)/(1-z^{-1})$	1	1
1	$A^1(z)/(1-z^{-1})^2$	z^{-1}	$1/z$
2	$A^2(z)/(1-z^{-1})^3$	$z^{-1}+z^{-2}$	$(1+z^{-1})/(2z)$
3	$A^3(z)/(1-z^{-1})^4$	$z^{-1}+4z^{-2}+z^{-3}$	$(z+4z+z^{-1})/(6z^2)$
4	$A^4(z)/(1-z^{-1})^5$	$(z^{-1}+z^{-2})(1+10z^{-1}+z^{-2})$	$(1+z^{-1})(z+10+z^{-1})/(24z^2)$
5	$A^5(z)/(1-z^{-1})^6$	$z^{-1}+26z^{-2}+66z^{-3}+26z^{-4}+z^{-5}$	$(z^2+26z+66+26z^{-1}+z^{-2})/(120z^3)$

3.3 İŞARETİN GÖSTERİLİŞİ VE İNTERPOLASYON

Bu bölümde genişleme katsayılarını belirlemek ve B-spline katsayılarından yararlanarak işareti yeniden elde etmek için hızlı çözümler araştırılacaktır. Bu hızlı çözümler rekürsif filteler üzerine kurulmuştur.

3.3.1 Doğrudan B-Spline Transformu

$k=-\infty.....+\infty$ aralığında tanımlanmış ayrık bir sinyal için $\phi^n(k)$ interolasyon fonksiyonunu bulmak için

$$f(k)=\phi^n(k)=\sum_{i=-\infty}^{+\infty} c(i)b_1{}^n(k-i)$$

yazılır. Bu eşitlik bir konvolüsyon işlemini gösterir ve $f(k)= b_1^n * c(k)$ olarak yazılabilir. Burada $b_1^n(k)$ daha önceden n.dereceden dolaylı spline filtre olarak belirttiğimiz operatörün sonlu impuls yanıtıdır. Z transformlarını alarak

$$F(z)=B_1^n(z)C(z)$$

elde edilir.$B_1^n(z)$ bölüm 3.2'de tanımlanmıştır. Bu demklem gösteriyorki basit bir ters filtreleme ile {c(k)} spline katsayıları oluşturulabilir. $S^n(k)$ n.dereceden doğrudan spline filtre olarak adlandırılır ve transfer fonksiyonu aşağıdaki gibidir.

$$S^n(z)=B_1^n(z)^{-1} = 1/(\Sigma b_1^n(k) z^{-k})$$

$S^n(z)$,Rekürsif sonsuz uzunluklu impuls cevap filtresi(IIR) nin yanıtıdır.$B_1^n(z)$'in köklerini belirlemek ve bu köklerin birim çember içinde kalıp kalmadığını görmek çok önemlidir. Tablo 3.2'de n=0'dan 5'e kadar doğrudan ve dolaylı B-spline filtreler için transfer fonksiyonları ve kutuplar gösterilmiştir. Tablo 3.2'de gösterilen operatörlerle çalışırken sadece tek dereceli filtrelerin düzgün olduğu görülmüştür. n'in çift olduğu derecelerde, mevcut yaklaşımı uygunsuz kılan z=1' de bir kutup oluşmaktadır. Ancak örneklenmiş Çift dereceli B-splinelarda 1/2 faz ötelemesi ile bu problem aşılmıştır.

Tablo 3.2 Doğrudan ve Dolaylı B-spline filtreler için Transfer fonksiyonları

n	Dolaylı B-spline filtre	Doğrudan B-spline filtre	Kutup
0	1	1	-
1	1	1	-
2	$(z+6+z^{-1})/8$	$8/(z+6+z^{-1})$	$\alpha1=\sqrt{8}-2$ $\alpha2=1/\alpha1$
3	$(z+4+z^{-1})/6$	$6/(z+4+z^{-1})$	$\alpha1=\sqrt{3}-2$ $\alpha2=1/\alpha1$
4	$(z^2+76z+230+76z^{-1}+z^{-2})/384$	$384/(z^2+76z+230+76z^{-1}+z^{-2})$	$\alpha1=-0.01372$ $\alpha2=-0.36134$ $\alpha3=1/\alpha1, \alpha4=1/\alpha2$
5	$(z^2+26z+66+26z^{-1}+z^{-2})/120$	$120/(z^2+26z+66+26z^{-1}+z^{-2})$	$\alpha1=-0.04309$ $\alpha2=-0.43057$ $\alpha3=1/\alpha1, \alpha4=1/\alpha2$

3.3.2 Dolaylı B-Spline Transformu

Dolaylı B-Spline transformu, B-Spline'ları kullanarak ayrık işaret değerlerinin yeniden oluşturulması işini yapar. Pek çok uygulamada, bir işareti daha yüksek örnekleme oranında gösterme işi, interpolasyon işleminin özel bir şekli olarak yorumlanır. Dolaylı transformasyon veya yeniden oluşturma işleminde m genişletme faktörü olarak kullanılır ve m'in tamsayı olması şartı vardır. Bu işlem m genişletme faktörü ile büyütme olarak da yorumlanabilir. m faktörü ile

interpolasyon veya işareti yeniden oluşturma işlemi aşağıdaki gibi yazılabilir.

$$f(k') = \phi^n(k'/m) = \sum_{i=-\infty}^{+\infty} c(i) b_m^n(k-im) \qquad (3.8)$$

f(k) işareti için m faktörü ile örnek oranını arttırma (up-sampling) işlemi sonucunda $\{[f]\Uparrow m(k')\}$ adını verdiğimiz bir işaret üretilir ve buda aşağıdaki gibidir.

$$\{[f]\Uparrow m(k') = \begin{cases} f(k) & k'=mk \\ 0 & \text{diğer durumlarda} \end{cases} \Longleftrightarrow F(z^m) \qquad (3.9)$$

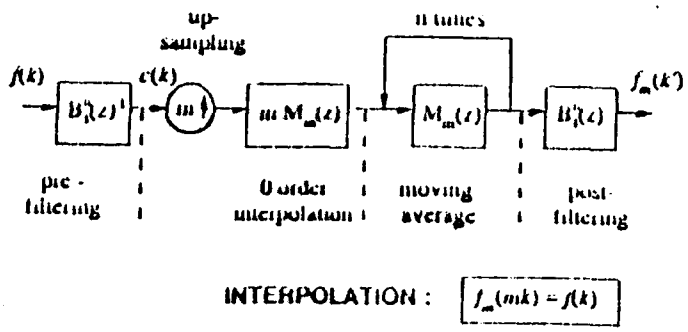
$[c]\Uparrow m(k')$ nü denklem 3.8'de kullanarak $f_m(k')=b_m^{n*}[c]\Uparrow m(k')$ yazılabilir. Bu eşitlik için z transformu alınırsa;

$$F_m(z) = B_1^n(z)\,(1/m^n)\,(B_m^0(z))^{n+1} C(z^m) \qquad (3.10)$$

olur. Buradan anlaşılacağı gibi işaret interpolasyonu, m boyutlu kayan ortalama (moving average) filtresinin n+1 defa ardarda kullanılması ile gerçekleştirilebilir.

3.4 TOPLU ÇÖZÜM SİSTEMİ

Yukarıda anlatılan işlemlerin gerçekleştirildiği sistem blok diyagramlar olarak Şekil 3.1'de gösterilmektedir. Şekilden de görüldüğü gibi spline katsayıları doğrudan B-spline transformasyonu ile oluşturulur. Daha sonraki üç işlem, m genişletme faktörü ile örnekleme oranını arttırma (up-sampling), m ile çarpma ve m boyutlu kayan ortalama filtresidir. Bu işaret kayan ortalama filtresinin $(M_m(z)=B_m^0(z)/m)$ n defa tekrarlanması ile düzeltilir. Son işlem ise dolaylı B-spline transformasyonudur.



Şekil 3.1 n.dereceden, m genişleme faktörlü B-spline interpolasyon blok diyagramı

Şekilde gösterilen işlemlerin tamamı bir konvolüsyon denklemi ile ifade edilebilir.

$$f_m(k')=b_m^{n*}[s^n]\uparrow m*[f]\uparrow m(k')=h_m^{n*}[f]\uparrow m(k')$$

$h_m^n(k')=b_m^{n*}[s^n]\uparrow m$ sistemin impulse yanıtıdır. Bu sistem için transfer fonksiyonu aşağıdaki gibidir.

$$H_m^n(z)=B_m^n(z)/B_1^n(z^m)=(1/m^n)B_1^n(z^m)\,(B_m^0(z))^{n+1}/B_1^n(z^m)$$

3.5 CUBIC B-SPLINE FİLTRE

Yukarıda geliştirilen çözümlerin sonuçlarını göstermek amacıyla doğrudan cubic B-spline filtre tasarımını inceleyeceğiz. Simetrik cubic B-spline z transformu aşağıdaki gibidir.

$$B_1^3(z)=(z+4+z^{-1})/6 \qquad (3.11)$$

Doğrudan B-spline filtre (ters filtre) için transfer fonksiyonu;

$$S^3(z)=6/(z+4+z^{-1})=-6a/((1-\alpha z^{-1})(1-\alpha z)) \qquad (3.12)$$

şeklinde yazılabilir. $\alpha=\sqrt{3}-2$ mutlak değer olarak en küçük kökü göstermektedir. $S^3(z)$, sabit simetrik sonsuz impuls yanıtlı (IIR) filtrenin transfer fonksiyonudur ve aşağıdaki gibi ayrıştırılabilir.

$$S^3(z)=(-6\alpha/(1-\alpha^2))(1/(1-\alpha z^{-1})+1/(1-\alpha z)) \qquad (3.13)$$

İmpuls yanıtı ise;

$$S^3(z)=(-6\alpha/(1-\alpha^2))\alpha^k$$

simetrik artan ve değişken işaretli üstel bir fonksiyondur. Bu filtre denklem 3.12 veya 3.13'ü kullanarak gerçekleştirilebilir. Bu durumda rekürsif filtre denklemleri;

$$c^+(k)=f(k)+b_1c^+(k-1) \quad (k=2,...K)$$
$$c^-(k)=f(k)+b_1c^-(k+1) \quad (k=K-1, ...1)$$
$$c(k) =b_0(c^+(k)+c^-(k)-f(k))$$

olur. Burada $b_0=-6\alpha/(1-\alpha^2)$ ve $b_1=\alpha=-0.2679$ dur.Ek olarak bazı başlangıç koşullarınıda göstermek gerekir.

$$c^+(1)= \sum_{k=1}^{k_0} \alpha^{k-1} f(k)$$

$$c^-(K)=c^+(K)$$

dır.

4. SONUÇ

Bu projede ana hedef, görüntü işlemede interpolasyon amacı ile de kullanılabilen B-spline fonksiyonlarını incelemek ve spline katsayılarının hesaplanmasında kullanılan algoritmayı, sonucu kötü etkilememek şartı ile, iyileştirmeye çalışmaktır. Bu amaçla bazı uygulamalarda spline katsayılarını hesaplamak yerine örnek değerleri matrisi bu katsayılar matrisi yerine kullanılmıştır. Fakat sonuç iyi olmamıştır. Yine alternatif bir çözüm olarak Cubic konvolüsyon interpolasyon geliştirilmiştir. Fakat cubic B-spline , cubic konvolüsyondan daha iyi sonuç vermektedir. Bu tez çalışmasında Bölüm 2'de verilen çözümleri, görüntü matrisini 7x7 ve 9x9 pixel boyutlarında parçalar halinde kullandık. Her pencere için ortadaki nokta ile ilgili ara değeri hesaplayarak pencereyi öteledik. Bu yöntemle, sonucu kabul edilebilir bir aralıkta olmak şartı ile işlem karmaşıklığı indirgenmeye çalışılmıştır. Deneysel görüntü sonuçları Bölüm 2'de Şekil 2.10'da gösterilmiştir.

EK-1

PCMatLab KAYNAK PROGRAMLARI

```
%            PCMatLab                    %
%     Tek boyutlu isaretlerde            %
% Lineer     B-Spline    interpolasyon %

format long;
e=zeros(17);
e(1,1)=1;

for j=2: 17
    e(j,j)=1;
end

x1=0: .125 : 2; f=sin(pi*x1);
f=f';
delta=.125;
a=(1/delta);
e=a*e;
c=inv(e)*f;
x=0: .0625 : 2; fb=sin(pi*x);

for i=1:16
     ck=c(i);
     fi(i)=a*(-c(i)*.5+c(i)+c(i+1)*.5);
     fb(i*2)=fi(i);
     ff(i)=sin(pi*(.125*(i-1)+.0625));
     fark(i)=ff(i)-fi(i);
end
fark(17)=0;
!del li17.met;
plot(x,fb,'b');
hold;
plot(x,fb,'o');
plot(x1,f,'r');
plot(x1,f,'+');
pause;

meta li17;
plot(x1,fark);
meta;
```

```
%            PCMatLab                    %
%     Tek boyutlu isaretlerde            %
% Quadratic    B-Spline   interpolasyon %

format long;
e=zeros(17);
e(1,1)=6;
e(1,2)=1;
for j=2: 16
  e(j,(j-1))=1;
  e(j,j)=6;
  e(j,(j+1))=1;
end
e(17,16)=1;
e(17,17)=6;
x=0: .125 : 2; f=sin(pi*x);
f=f';
delta=.125;
a=(1/(8*delta));
e=a*e;
x=.5;
x2=x^2;

omega=[ 4 -8  4
        -4  0  4
        1  6  1 ];

c=inv(e)*f;
x=0: .0625 : 2; fb=sin(pi*x);

for i=2:16
     ck=[c(i-1),c(i),c(i+1)];
     ck=ck';
     b=omega*ck;
     fi(i)=a*(b(1)*x2+b(2)*.5+b(3));
     fb(i*2)=fi(i);
     ff(i)=sin(pi*(.125*(i-1)+.0625));
     fark(i)=ff(i)-fi(i);
end
!del qu17.met;
plot(x,fb,'o');
hold;
plot(x,fb);

x=0: .125 :2  ;
plot(x,f,'w');
plot(x,f,'+');
meta qu17

pause;
hold;
fark(17)=0;
plot(x,fark);
meta;
```

```
%           PCMatLab                    %
%    Tek boyutlu isaretlerde            %
% Cubic    B-Spline   interpolasyon %

format long;
e=zeros(17);
e(1,1)=4;
e(1,2)=1;
for j=2: 16
  e(j,(j-1))=1;
  e(j,j)=4;
  e(j,(j+1))=1;
end
e(17,16)=1;
e(17,17)=4;

x=0: .125 : 2; f=sin(pi*x);
f=f'
delta=.125;
a=(1/(6*delta));
e=a*e;

x=.5;
x3=x^3;
x2=x^2;

omega=[-1 3 -3 1
        3 -6 3 0
        -3 0 3 0
        1 4 1 0];


c=inv(e)*f;
x=0: .0625 : 2; fb=sin(pi*x);

for i=2:15
     ck=[c(i-1),c(i),c(i+1),c(i+2)];
     ck=ck';
     b=omega*ck;
     fi(i)=a*( b(1)*x3 + b(2)*x2 + b(3)*.5 +b(4));
     fb(i*2)=fi(i);
     ff(i)=sin(pi*(.125*(i-1)+.0625));
     fark(i)=ff(i)-fi(i);
end
x1=0: .125 : 2
!del cu17.met;
plot(x,fb,'o');
hold;

plot(x,fb)
meta cu17
plot(x1,f,'w');
plot(x1,f,'+');
meta
```

```
hold

fark(17)=0;
plot(x1,fark);
meta cu17;
fark(16)=0;
plot(x1,fark);
meta;
```

```
%      PCMatLab
% Iki boyutlu isaretlerde
%    Cubic B-spline
% Her nokta icin iki yeni nokta bulur.
format long;
ekr1;

e=zeros(50);
e(1,1)=4; e(1,2)=1;
for j=2: 49
  e(j,(j-1))=1;
  e(j,j)=4;
  e(j,(j+1))=1;
end
e(50,49)=1; e(50,50)=4;

delta=3;
deltk=delta^2;

e=(1/(6*delta))*e;
ei=inv(e);
c=ei*(sdata*ei);

clear sdata;
clear e;
clear ei;
pack;

x=1/3;
x3=x^3;
x2=x^2;

xb=2/3;
xb3=xb^3;
xb2=xb^2;

y=1/3;
y3=(3+y)^3;
y2=(2+y)^3;
y1=(1+y)^3;
yk=y^3;

yb=2/3;
yb3=(3+yb)^3;
yb2=(2+yb)^3;
yb1=(1+yb)^3;
ybk=yb^3;

a=(1/(36*deltk));
omega=[-1 3 -3 1
        3 -6 3 0
        -3 0 3 0
        1 4 1 0];
for 1=2: 48
```

```
for k=2: 48

    j=1;
    ck=[c((k-1),(l-1)) , c(k,(l-1)) , c((k+1),(l-1)),
        c((k+2),(l-1))] ;
    ck=ck';
    b=omega*ck;
    fi(j)=b(1)*x3 +b(2)*x2 +b(3)*x +b(4) ;
    fbi(j)=b(1)*xb3 +b(2)*xb2 +b(3)*xb +b(4) ;
    fy(j)=b(4);

    j=j+1;
    ck=[c((k-1),l) , c(k,l) , c((k+1),l), c((k+2),l)] ;
    ck=ck';
    b=omega*ck;
    fi(j)=b(1)*x3 +b(2)*x2 +b(3)*x +b(4) ;
    fy(j)=b(4);

    j=j+1;
    ck=[c((k-1),(l+1)) , c(k,(l+1)) , c((k+1),(l+1)),
       c((k+2),(l+1))] ;
    ck=ck';
    b=omega*ck;
    fi(j)=b(1)*x3 +b(2)*x2 +b(3)*x +b(4) ;
    fbi(j)=b(1)*xb3 +b(2)*xb2 +b(3)*xb +b(4) ;
    fy(j)=b(4);

    j=j+1;
    ck=[c((k-1),(l+2)) , c(k,(l+2)) , c((k+1),(l+2)),
        c((k+2),(l+2))] ;
    ck=ck';
    b=omega*ck;
    fi(j)=b(1)*x3 +b(2)*x2 +b(3)*x +b(4) ;
    fbi(j)=b(1)*xb3 +b(2)*xb2 +b(3)*xb +b(4) ;
    fy(j)=b(4);

    fix1y0(k,l)=a*(fi(1)+ 4*fi(2)+ fi(3));
    fix2y0(k,l)=a*(fbi(1)+ 4*fbi(2) +fbi(3));

    fix0y1(k,l)=a*((fy(1)*(y3 -4*y2 +6*y1 -4*yk))+
                (fy(2)*(y2 -4*y1+ 6*yk))+ (fy(3)*(y1-4*yk))
                +(fy(4)*yk));
    fix0y2(k,l)=a*((fy(1)*(yb3 -4*yb2 +6*yb1 -4*ybk))+
                (fy(2)*(yb2 -4*yb1+ 6*ybk))+ (fy(3)*(yb1
                -4*ybk)) +(fy(4)*ybk));

    fix1y1(k,l)=a*((fi(1)*(y3 -4*y2 +6*y1 -4*yk))+
                (fi(2)*(y2 -4*y1+ 6*yk))+ (fi(3)*(y1-4*yk))
                +(fi(4)*yk));
    fix1y2(k,l)=a*((fi(1)*(yb3 -4*yb2 +6*yb1 -4*ybk))+
                (fi(2)*(yb2 -4*yb1+ 6*ybk))+ (fi(3)*(yb1
                -4*ybk)) +(fi(4)*ybk));

    fix2y1(k,l)=a*((fbi(1)*(y3 -4*y2 +6*y1 -4*yk))+
```

```
                (fbi(2)*(y2 -4*y1+ 6*yk))+ (fbi(3)*(y1
                -4*yk)) +(fbi(4)*yk));
    fix2y2(k,l)=a*((fbi(1)*(yb3 -4*yb2 +6*yb1 -4*ybk))+
                (fbi(2)*(yb2 -4*yb1+ 6*ybk))+ (fbi(3)*(yb1
                -4*ybk)) +(fbi(4)*ybk));

  end
end

clear c;
pack;

format short;
fix1y0=round(fix1y0);
fix1y0=abs(fix1y0);

fix2y0=round(fix2y0);
fix2y0=abs(fix2y0);

fix0y1=round(fix0y1);
fix0y1=abs(fix0y1);

fix1y1=round(fix1y1);
fix1y1=abs(fix1y1);

fix2y1=round(fix2y1);
fix2y1=abs(fix2y1);

fix0y2=round(fix0y2);
fix0y2=abs(fix0y2);

fix1y2=round(fix1y2);
fix1y2=abs(fix1y2);

fix2y2=round(fix2y2);
fix2y2=abs(fix2y2);



!del ekx1y0.txt;
!del ekx2y0.txt;
!del ekx0y1.txt;
!del ekx1y1.txt;
!del ekx2y1.txt;
!del ekx0y2.txt;
!del ekx1y2.txt;
!del ekx2y2.txt;

for i=1: 48
 for k=1: 48
   fprintf('ekx1y0.txt','%f\n',fix1y0(i,k));
   fprintf('ekx2y0.txt','%f\n',fix2y0(i,k));
   fprintf('ekx0y1.txt','%f\n',fix0y1(i,k));
   fprintf('ekx1y1.txt','%f\n',fix1y1(i,k));
```

```
    fprintf('ekx2y1.txt','%f\n',fix2y1(i,k));
    fprintf('ekx0y2.txt','%f\n',fix0y2(i,k));
    fprintf('ekx1y2.txt','%f\n',fix1y2(i,k));
    fprintf('ekx2y2.txt','%f\n',fix2y2(i,k));
 end
end
```

```
%       PCMatLab
% Iki boyutlu isaretlerde
%   Cubic B-spline
% Her nokta icin bir yeni nokta bulur.
format long;
ekr1;

e=zeros(50);
e(1,1)=4; e(1,2)=1;
for j=2: 49
  e(j,(j-1))=1;
  e(j,j)=4;
  e(j,(j+1))=1;
end
e(50,49)=1; e(50,50)=4;

delta=1;
deltk=delta^2;

e=(1/(6*delta))*e;
ei=inv(e);
c=ei*(sdata*ei);

x=.5;
x3=x^3;
x2=x^2;

y=.5;
y3=(3+y)^3;
y2=(2+y)^3;
y1=(1+y)^3;
yk=y^3;

a=(1/(36*deltk));
omega=[-1 3 -3 1
        3 -6 3 0
        -3 0 3 0
        1 4 1 0];
for l=2: 48
 for k=2: 48

    j=1;
    ck=[c((k-1),(l-1)) , c(k,(l-1)) , c((k+1),(l-1)),
       c((k+2),(l-1))] ;
    ck=ck';
    b=omega*ck;
    fi(j)=b(1)*x3 +b(2)*x2 +b(3)*x +b(4) ;
    fy(j)=b(4);

    j=j+1;
    ck=[c((k-1),l) , c(k,l) , c((k+1),l), c((k+2),l)] ;
    ck=ck';
    b=omega*ck;
    fi(j)=b(1)*x3 +b(2)*x2 +b(3)*x +b(4) ;
```

```
    fy(j)=b(4);

    j=j+1;
    ck=[c((k-1),(l+1)) , c(k,(l+1)) , c((k+1),(l+1)),
       c((k+2),(l+1))] ;
    ck=ck';
    b=omega*ck;
    fi(j)=b(1)*x3 +b(2)*x2 +b(3)*x +b(4) ;
    fy(j)=b(4);

    j=j+1;
    ck=[c((k-1),(l+2)) , c(k,(l+2)) , c((k+1),(l+2)),
       c((k+2),(l+2))] ;
    ck=ck';
    b=omega*ck;
    fi(j)=b(1)*x3 +b(2)*x2 +b(3)*x +b(4) ;
    fy(j)=b(4);

    fix(k,l)=a*(fi(1)+ 4*fi(2)+ fi(3));
    fiy(k,l)=a*((fi(1)*(y3 -4*y2 +6*y1 -4*yk))+ (fi(2)*(y2
            -4*y1+ 6*yk))+ (fi(3)*(y1-4*yk)) +(fi(4)*yk));
    fiy2(k,l)=a*((fy(1)*(y3 -4*y2 +6*y1 -4*yk))+ (fy(2)*(y2
             -4*y1+ 6*yk))+ (fy(3)*(y1-4*yk)) +(fy(4)*yk));
  end
end

clear e;
clear c;
clear ei;
clear sdata;
pack;

format short;
fix=round(fix);
fiy=round(fiy);
fiy2=round(fiy2);


for i=1:48
  for j=1:48
    if  fix(i,j)<0
        fix(i,j)=0;
    end

    if  fiy(i,j)<0
        fiy(i,j)=0;
    end

    if  fiy2(i,j)<0
        fiy2(i,j)=0;
    end


  end
```

```
end

!del ek50x.txt;
!del ek50y.txt;
!del ek50y2.txt;

for i=1: 48
 for k=1: 48
   fprintf('ek50x.txt','%f\n',fix(i,k));
   fprintf('ek50y.txt','%f\n',fiy(i,k));
   fprintf('ek50y2.txt','%f\n',fiy2(i,k));
 end
end
```

EK-2

C ++ KAYNAK PROGRAMLARI

```
/* Ekranda f=sin(0.5r2) fonksiyonunu gri seviyelerle cizer*/
/* Olusan sekli PCMatLab icin matris seklinde */
/* bir file'a yazar.*/

#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <conio.h>

int main(void)
{

/* select a driver and mode that supports the use */
/* of the setrgbpalette function.                 */
 int gdriver=VGA , gmode=VGAHI , errorcode;
 struct palettetype pal;
 int N,i, r,b,inbuf ;
 char bo=' ',cr='\n';
 double j,k=2.0;
 FILE  *ekfile;


 /* grab a copy of the palette */
getpalette(&pal);
  gdriver = IBM8514;
  gmode = IBM8514HI;

  initgraph(&gdriver,&gmode,"");
  for(N=0;N<256;N++)
      setrgbpalette(N,N,N,N);

 cleardevice();


/* draw sin(.5r^2) function */
for (r=1; r<350; r++)
   {
     j=r*3;
     i=128+128*sin(.5*pow((j/100.0),2.0));
     setfillstyle(0,i);
     setcolor(i);
     circle(0,0,r);
     circle(1,1,r);
   }

/* ekrandaki resim file'a yazilir */
  /* open file, testing for success */
 if ((ekfile=fopen("ekr1.m","w+b"))==NULL){
    printf("Error opening text file for writting\n");
    exit(0);
    }
```

```
 /*write some bytes to the file */
 for (i=0; i<50; i++)
 {
   for (k=0; k<50; k++)
   {
    inbuf=getpixel(k*3,i*3);
    fprintf(ekfile,"%d",inbuf);
    fprintf(ekfile,"%c",bo);
    }
    fprintf(ekfile,"%c",cr);
 }

 fclose(ekfile);

getch();
 closegraph();
 return 0;
 }
```

```
/* PCMatLab in Cubic B-spline interpolasyonla buldugu */
/* yeni degerleri ekranda yerine yazar.               */

#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <conio.h>

int main(void)
{

/* select a driver and mode that supports the use */
/* of the setrgbpalette function.                 */
 int gdriver=VGA, gmode=VGAHI, errorcode;
 struct palettetype pal;
 int i,N,ii,r,b,inbuf,bb ;
 float j, k=2.0;
 FILE  *ekfile;
 FILE *temp;


  /* grab a copy of the palette */
getpalette(&pal);
  gdriver = IBM8514;
  gmode = IBM8514HI;

  initgraph(&gdriver,&gmode,"");
  for(N=0;N<256;N++)
      setrgbpalette(N,N,N,N);

 cleardevice();



/* draw sin(.5r^2) function */
for (r=1; r<250; r++)
   {
     j=r*3;
     i=128+128*sin(.5*pow((j/100.0),k));
     setfillstyle(0,i);
     setcolor(i);
     circle(0,0,r);
     circle(1,1,r);
   }

setfillstyle(0,0);
bar(150,0,640,480);

if ((ekfile=fopen("ekx0y1.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);
    }
for (i=0; i<48; i++)
```

```
 {
   ii=i*3;
   for (b=0; b<48; b++)
   {
    bb=b*3;
    inbuf=getpixel(bb,ii);
    putpixel(150+bb,ii,inbuf);
    putpixel(150+bb,ii+200,inbuf);
    putpixel(150+bb,ii+201,inbuf);
    putpixel(150+bb,ii+202,inbuf);
    putpixel(151+bb,ii+201,inbuf);
    putpixel(151+bb,ii+200,inbuf);
    putpixel(151+bb,ii+202,inbuf);
    putpixel(152+bb,ii+200,inbuf);
    putpixel(152+bb,ii+201,inbuf);
    putpixel(152+bb,ii+202,inbuf);
    fscanf(ekfile,"%f",&k);
    inbuf=k;
    if (inbuf>255) inbuf=255;
    putpixel(151+bb,ii,inbuf);
    }
 }
fclose(ekfile);

if ((ekfile=fopen("ekx0y2.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);

    }
for (i=0; i<48; i++)
 {
   ii=i*3;
   for (b=0; b<48; b++)
   {
    bb=b*3;
    fscanf(ekfile,"%f",&k);
    inbuf=k;
    if (inbuf>255) inbuf=255;
    putpixel(152+bb,ii,inbuf);
    }
 }
fclose(ekfile);

if ((ekfile=fopen("ekx1y0.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);
   }
for (i=1; i<49; i++)
 {
   ii=(i*3)-2;
   for (b=0; b<48; b++)
   {
    bb=b*3;
    fscanf(ekfile,"%f",&k);
```

```
    inbuf=k;
    if (inbuf>255) inbuf=255;
    putpixel(150+bb,ii,inbuf);
    }
 }
fclose(ekfile);

if ((ekfile=fopen("ekx1y1.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);
   }
for (i=1; i<49; i++)
 {
   ii=(i*3)-2;
   for (b=0; b<48; b++)
   {
    bb=b*3;
    fscanf(ekfile,"%f",&k);
    inbuf=k;
    if (inbuf>255) inbuf=255;
    putpixel(151+bb,ii,inbuf);
    }
 }
fclose(ekfile);

if ((ekfile=fopen("ekx1y2.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);
   }
for (i=1; i<49; i++)
 {
   ii=(i*3)-2;
   for (b=0; b<48; b++)
   {
    bb=b*3;
    fscanf(ekfile,"%f",&k);
    inbuf=k;
    if (inbuf>255) inbuf=255;
    putpixel(152+bb,ii,inbuf);
    }
 }
fclose(ekfile);

if ((ekfile=fopen("ekx2y0.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);
   }
for (i=1; i<49; i++)
 {
   ii=(i*3)-1;
   for (b=0; b<48; b++)
   {
    bb=b*3;
    fscanf(ekfile,"%f",&k);
```

```
    inbuf=k;
    if (inbuf>255) inbuf=255;
    putpixel(150+bb,ii,inbuf);
    }
 }
fclose(ekfile);

if ((ekfile=fopen("ekx2y1.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);
   }
for (i=1; i<49; i++)
 {
   ii=(i*3)-1;
   for (b=0; b<48; b++)
   {
    bb=b*3;
    fscanf(ekfile,"%f",&k);
    inbuf=k;
    if (inbuf>255) inbuf=255;
    putpixel(151+bb,ii,inbuf);
    }
 }
fclose(ekfile);

if ((ekfile=fopen("ekx2y2.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);
   }
for (i=1; i<49; i++)
 {
   ii=(i*3)-1;
   for (b=0; b<48; b++)
   {
    bb=b*3;
    fscanf(ekfile,"%f",&k);
    inbuf=k;
    if (inbuf>255) inbuf=255;
    putpixel(152+bb,ii,inbuf);
    }
 }
fclose(ekfile);

for (i=0; i<148; i++)
{
  for (b=0; b<148; b++)
  {
  inbuf=abs(getpixel(i,b)-getpixel(150+i,b));
  putpixel(300+i,b,inbuf);
  inbuf=abs(getpixel(i,b)-getpixel(150+i,200+b));
  putpixel(300+i,200+b,inbuf);
  }
}
```

```
getch();
 closegraph();
 return 0;
 }
```

```
/* Tif File formatini çözerek Cubic B_spline */
/* interpolasyonla hesaplanan yeni degerleri ekrana yazar.*/

#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <conio.h>

int main(void)
{
   int konvert(char);
/* select a driver and mode that supports the use */
/* of the setrgbpalette function.                 */
 int gdriver=VGA, gmode=VGAHI, errorcode;
 struct palettetype pal;
 int i,ii,r,b,inbuf,bb ;
 int N,sayx,sayy;
 char ch1,ch2,inbu[435];
 float j, k=2.0;
 FILE  *ekfile;

 /* grab a copy of the palette */
getpalette(&pal);
  gdriver = IBM8514;
  gmode = IBM8514HI;

  initgraph(&gdriver,&gmode,"");
  for(N=0;N<256;N++)
      setrgbpalette(N,N,N,N);

 cleardevice();

    if ((ekfile=fopen("arap4.tif","r+b"))==NULL)
 {
    printf("Error opening text file for writting\n");
    exit(0);
    }
 fseek(ekfile,30,SEEK_SET);
 fscanf(ekfile,"%c",&ch1);
 fscanf(ekfile,"%c",&ch2);
 sayx=ch2*256+konvert(ch1);
 fseek(ekfile,10,SEEK_CUR);
 fscanf(ekfile,"%c",&ch1);
 fscanf(ekfile,"%c",&ch2);
 sayy=ch2*256+konvert(ch1);


 fseek(ekfile,866,SEEK_SET);
 for (i=0; i<sayy; i++)
  {
  fread(inbu,sizeof(inbu),1,ekfile);
  for (b=0; b<sayx ; b++)
  {
```

```
      ch1=inbu[b];
     putpixel(b,i,                putpixel(b,i,ch1);
   }
  }

 /*close the file*/
 fclose(ekfile);

setfillstyle(0,0);
bar(150,0,640,480);

if ((ekfile=fopen("ary2.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);
     }
for (i=0; i<48; i++)
 {
   ii=i*2;
   for (b=0; b<48; b++)
   {
    bb=b*2;
    inbuf=getpixel(bb,ii);
    putpixel(150+bb,ii,inbuf);
    putpixel(150+bb,ii+200,inbuf);
    putpixel(150+bb,ii+201,inbuf);
    putpixel(151+bb,ii+201,inbuf);
    putpixel(151+bb,ii+200,inbuf);
    fscanf(ekfile,"%f",&k);
    inbuf=k;
    if  (inbuf>255) inbuf=255;
    putpixel(151+bb,ii,inbuf);
    }
 }
fclose(ekfile);

if ((ekfile=fopen("arx.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);

     }
for (i=1; i<49; i++)
 {
   ii=(i*2)-1;
   for (b=0; b<48; b++)
   {
    bb=b*2;
    fscanf(ekfile,"%f",&k);
    inbuf=k;
    if  (inbuf>255) inbuf=255;
    putpixel(150+bb,ii,inbuf);
    }
 }
fclose(ekfile);
```

```
if ((ekfile=fopen("ary.txt","rt"))==NULL){
outtext("Error opening text file for writting\n");
exit(0);
   }
for (i=1; i<49; i++)
 {
   ii=(i*2)-1;
   for (b=0; b<48; b++)
   {
    bb=b*2;
    fscanf(ekfile,"%f",&k);
    inbuf=k;
    if  (inbuf>255) inbuf=255;
    putpixel(151+bb,ii,inbuf);
    }
 }
fclose(ekfile);

for (i=0; i<98; i++)
{
  for (b=0; b<98; b++)
  {
  inbuf=abs(getpixel(i,b)-getpixel(150+i,b));
  putpixel(300+i,b,inbuf);
  inbuf=abs(getpixel(i,b)-getpixel(150+i,200+b));
  putpixel(300+i,200+b,inbuf);
  }
}

getch();
 closegraph();
 return 0;
 }

int konvert(char a1)
{
  int sayi,temp;
  sayi=a1 & 15;
  a1=a1 >> 4;
  a1=a1 & 15;
  sayi=sayi+a1*16;
  return sayi;
}
```

ÖZGEÇMİŞ

Songül ALBAYRAK, 1968 yılında İstanbul'da doğdu. Liseyi K.Maltepe Lisesi'nde bitirdi. 1989 yılında Yıldız Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümünden mezun oldu. 1990 yılında Yıldız Üniversitesi Mühendislik Fakültesi Bilgisayar Bilimleri ve Mühendisliği Bölümünde Yüksek Lisans çalışmasına başladı. Aynı dönemde bu bölümün Donanım Anabilim Dalında Araştırma Görevlisi olarak çalışmaya başlamıştır. Halen öğrenimine ve görevine devam etmektedir.

KAYNAKLAR

* Robert G.Keys, "Cubic Convolution Interpolation for Digital Image Processing"
  IEEE Transaction on Acoustic,Speech, and Signal Processing, Vol.ASSP-29 , No.6 , December 1981

* H.S.Hou ve H.C.Andrews, "Cubic Splines for Image Interpolation and Digital Filtering"
  IEEE Transaction on Acoustic,Speech, and Signal Processing, Vol.ASSP-29 , No.6 , December 1978

* E. Maeland,"On the Comparison of Interpolation Methods"
  IEEE Transaction on Medical Imaging,
  Vol.7, No.3, September 1988

* J.A.Parker, R.V.Kenyon ve D.E.Troxel, "Comparison of Interpolating Methods for Image Resampling"
  IEEE Transaction on Medical Imaging,
  Vol.MI-2, No.1, March 1983

* M.Unser, A.Aldroubi ve M.Eden, "Fast B-spline Transforms for Continuous Image Representation and Interpolation"
  IEEE Transaction on Pattern Analysis and Machine Intelligence,
  Vol.13, No.3, March 1991

* P.V.Sankar ve L.A.Ferrari, "Simple Algorithms and Architectures for B-spline Interpolation"
  IEEE Transaction on Pattern Analysis and Machine Intelligence,
  Vol.10, No.2, March 1988

* K.Toraichi, S.Yang, M.Kamada ve R.Mori, "Two-dimensional

Spline Interpolation for Image Reconstruction"
Pattern Recognition, Vol.21, No.3 1988

* M.Unser, A.Aldroubi ve M.Eden, "Cardinal Spline Filters: Stability and Convergence to the Ideal Sinc Interpolator"
Signal Processing, Vol.28 1992

* A.V.Oppenheim ve R.W. Schafer
Digital Signal Processing, Prentice-Hall, Inc. 1975

* John R.Rice
Numerical Methods, Software, and Analysis
McGraw Hill, 1983

* Tag Image File Format, Rev.4.0, 1987

* H.Schildt, C Made Easy, McGraw Hill

* Ziya Aktaş, Sayısal Çözümleme, Mart 1981

* Turbo C++ Users Guide

* MatLab Users Guide