ÇAĞRIŞIMLI HAFIZA YÖNTEMİYLE VERİ İŞLEME

Nihat Adar

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik ve Elektronik
Mühendisliği Ana Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır.

Danışman : Doç.Dr.Atila Barkana

Şubat-1988

Nihat Adar 'ın YÜKSEK LİSANS tezi olarak hazırladığı " Çağrışımlı Hafıza Yöntemiyle Veri İşleme " başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

27./2./1988

Başkan: .....
Prof.Dr. İmdat Kara

Üye : .....
Prof.Dr. Atalay Barkana

Üye : .....
Doç.Dr. Atila Barkana

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..2-3-1988..... gün ve .....169-22..... sayılı kararıyla onaylanmıştır.

Enstitü Müdürü
Prof.Dr. Rüstem Kaya

ÖZET

Bu çalışmanın amacı, çağrışımlı metodlarla veri işlenmesini gerçekleştirmektir. Bunun için de en gelişmiş çağrışımlı hafızaya sahip insan beyninin hafıza işlevinin çalışma prensibinden yola çıkılmış ve böylece geliştirilmiş olan bir matematiksel model ile çalışılmıştır. Matematiksel modelin çalışması bilgisayar ortamındaki simulasyonla incelenmiştir. Verilmiş olan resimler arasından seçilmiş bir anahtar resim önce bozulmuş ve sonra bilgisayar hafızasındaki diğer resimlerden tekrar derlenmiştir.

Bu sistem, resmin doğrudan bilgisayar hafızasından alınarak değil, hafızadaki resimlerin ortak katkısıyla tekrar oluşturulması şeklindedir. Matematiksel modelde, anahtar resim, aralarından tanınması istenilen (bilgisayar hafızasındaki) resimlerin bir lineer birleşimidir. Bu sayede bir kısmı bozulmuş veya tamamen silinmiş bir anahtar resmin diğer resimler arasından tanınması sözkonusu olabilmektedir.

Ayrıca resimler üzerine oluşturulmuş olan bu modelin diğer veri türleri üzerinde de etkin bir şekilde kullanılması mümkündür.

SUMMARY

The purpose of this work is to realize the data processing by associative methods. For this, the mathematical model for the operation of the human brain, which has the most developed associative memory, is used. The mathematical model is examined by computer simulation.

First, a key picture chosen from a set of pictures is deformed and then this picture is recreated from the pictures in computer's memory. This process was not accomplished by selecting the data of the picture directly from the memory, instead, the picture is created by the combination of the contributions of all the pictures in the memory. In the mathematical model, the key picture is a lineer combination of the pictures which are present in the memory. For this reason, a key picture which is deformed or erased partially or which is added some type of noise can be recognized.

The mathematical model can also be used with any other type of data besides picture data.

TEŞEKKÜR

Çalışmalarım esnasında araştırmadaki matematiksel modelin irdelenmesi konusunda destek olan,

Doç.Dr. Şahin Koçak 'a (A.Ü. Fen Bil. Fak. Matematik Bl.)

çalışmanın programlama ve yazımı esnasında yardımcı olan,

Araş.Grv. Selçuk Canbek 'e (A.Ü. Fen Bil. Fakültesi)

çalışmalarım süresince benden hiçbir yardımını esirgemeyen danışman hocam,

Doç.Dr. Atila Barkana 'ya (A.Ü.M.M.F. Elektrik ve Elektronik Müh. Bl.)

ve tezin yazımı esnasında emeği geçen tüm arkadaşlarıma

teşekkürlerimi bir borç bilirim.

İÇİNDEKİLER

İÇİNDEKİLER (devam)

Sayfa

ŞEKİLLER DİZİNİ

1. GİRİŞ

1.1. Giriş

Günümüz teknolojisindeki hızlı gelişme ile yüksek performanslı, büyük hafızalı bilgisayarlar insanlığın hizmetine sunulmuştur. Bunun paralelinde eskiden bir ütopya olan pek çok konu çözüme kavuşturulmaya başlamıştır. Böylece pek çok canlı organizmanın modellerinin geliştirilmesi başarılabilmiştir. Aynı şekilde daha on yıl öncesine kadar felsefi tartışmalardan öteye gidemeyen beyin konusundaki çalışmalar da artık bilimsel temeller üzerinde yükselmeye başlamıştır. İnsan beyninin en önemli fonksiyonlarından biri olan hafızanın bir modelinin geliştirilmesi konusunda da pek çok çalışmalar gerçekleştirilmiştir. Günümüz teknolojisinde her alanda olduğu gibi bu alandaki çalışmaların uzantıları da farklı alanlarda uygulama sahası bulabilmektedir.

1.2. Amaç

Aslında insan beyninin hafıza işlevinin çözümlenmesi konusundaki araştırmalar sonuçta göstermiştir ki; bir nesnenin değerlendirilip tanınması ile herhangi bir veri yığınının değerlendirilmesi büyük ölçüde eşdeğerdir. Bu durumda bir matematiksel model geliştirirken yapılacak olan haricen verilebilecek anahtar verinin; büyük bir veri yığını içinde olup olmadığını araştırmak olacaktır. Eğer anahtar veri ilgili veri yığını içinde var ise, "hangi veriye ne kadar benziyor? " sorusuna da cevap verebilmelidir.

Bu çalışmanın amacı da ; böyle matematiksel bir modelin geliştirilip, bilgisayar ortamında işlenmesini sağlamaktır. Bu tip matematiksel modeller daha önceden de geliştirilmiştir. Fakat büyük veri yığınlarından söz edilmeye başlayınca el ile hesaplanamayacak (hesaplansa bile bir anlamı kalmayacak)bir işlem yükünden bahsedilmektedir. Bu nedenle

teknolojinin istenilen seviyeye gelmesine kadar bu tip çalışmaları bir noktadan sonra durdurmak zorunda kalınmıştır.

1.3. Hedef

Yukarıda açıklanılan amaç doğrultusunda hedef, pek çok veri arasından, haricen verilecek bir anahtar verinin aranması veya başka bir deyişle hafızadaki veriler yardımı ile anahtar verinin tekrar oluşturulması olmaktadır. Burada veri yığını olarak bir vektör uzayından bahsedilmektedir. Bu vektör uzayı, kullanıcı tarafından tanımlanmış vektörlerden oluşmaktadır. Anahtar veri de,bu vektör uzayının elemanı olan herhangi bir vektördür. Bu vektörün elemanlarının bazılarının değeri bozularak da anahtar veri tanımlanabilir

1.4. Matematiksel Metodlar ve Notasyonlar

İzleyicinin ilgili matematiksel modeli daha rahat bir şekilde izleyebilmesi açısından, çalışmada kullanılan bazı matematiksel kavramlara kısaca değinilecektir.

4.1.1. Vektör uzayı kavramı

Çalışmada sâbit katsayılar küçük Yunan karakterleri ile, vektörler ve indisler aksi belirtilmediği sürece, küçük Latin karakterleri ile temsil edilmiştir.

Lineer vektör uzayları :

Lineer bir vektör uzayı (£) genellikle üzerinde vektörel işlemlerden toplama, skaler çarpma gibi işlemlerin yapılabildiği, vektörlerden oluşan elemanların oluşturduğu bir uzay olmaktadır. Eğer $x,y,z \in £$ vektörler ve σ,β skalar sayılar olursa aşağıdaki özellikler £ vektör uzayında tanımlıdır.

1) $x+y = y+x \in £$ — değişme özelliği

2) $\sigma*(x+y) = \sigma*x + \sigma*y \in R$

3) $(\sigma+\beta)*x = \sigma*x + \beta*x$ — dağılma özelliği

4) $(x+y)+z = x+(y+z)$

5) $(\sigma*\beta)*x = \sigma*(\beta*x)$ — birleşme özelliği

6) $x \in £$ için $x+0 = x$

7) 0 ve 1 skaler olmak üzere;
$0*x=0$ ve $1*x=x$ olur. (1.1)

Skaler ile çarpımda vektörün her bir elemanı skaler ile çarpılmaktadır.

İki vektörün skaler çarpımı ise, $x=(\beta_1,\beta_2,\ldots,\beta_n)$ ve $y=(\sigma_1,\sigma_2,\ldots,\sigma_n)$ vektörler olmak üzere,

$$\langle x,y\rangle = \beta_1\sigma_1+\beta_2\sigma_2+\ldots+\beta_n\sigma_n \qquad (1.2)$$

olarak tanımlanmaktadır.

Skaler çarpım aşağıdaki özellikleri sağlamak zorundadır.

1) $\langle x,y\rangle = \langle y,x\rangle$

2) $\langle \sigma x,y\rangle = \sigma\langle x,y\rangle$

3) $\langle x_1+x_2,y\rangle = \langle x_1,y\rangle+\langle x_2,y\rangle$

4) $\langle x,x\rangle \geq 0$ eşitlik yalnız $x=0$ için geçerlidir. (1.4)

İki vektör arasındaki uzaklık d(x,y) olarak gösterilecektir ve kartezyen koordinat sisteminde $x=(\beta_1,\beta_2,\ldots,\beta_n)$ ve $y=(\sigma_1,\sigma_2,\ldots,\sigma_n)$ olmak üzere;

$$d(x,y)=\sqrt{(\beta_1-\sigma_1)^2+(\beta_2-\sigma_2)^2+\ldots+(\beta_n-\sigma_n)^2} \qquad (1.4)$$

olarak tanımlanmakta ve **Euclidian uzaklık** (Euclidian distance) olarak adlandırılmaktadır.

Uzaklık fonksiyonunun seçiminde aşağıdaki şartlar sağlanmak zorundadır.

1) $d(x,y) \geq 0$, eşitlik yalnız x=y olunca geçerlidir.

2) $d(x,y) = d(y,x)$

3) $d(x,y) \leq d(x,z) + d(z,y)$ (1.5)

Diğer bir uzaklık örneğide **Hamming uzaklığı**(Hamming distance) olarak adlandırılan ikili (binary) vektörler için tanımlı olan uzaklık fonksiyonudur. Hamming uzaklık fonksiyonu iki adet ikili vektörün kaç tane farklı elemanı olduğunu gösterir. Denklem 1.5 'teki 1. ve 3. şartlar Hamming uzaklık fonksiyonu için de geçerlidir.

Bir vektörün normu farklı şekillerde tanımlanabilir. Yalnız norm fonksiyonunun tanımında aşağıdaki kurallar sağlanmak zorundadır.

1) norm(x) ≥ 0 , eşitlik yalnız x=0 için geçerlidir.

2) $norm(\sigma x) = |\sigma| norm(x)$ burada $|\sigma|$; σ'ın mutlak değerini gösterir.

3) $norm(x_1+x_2) \leq norm(x_1)+norm(x_2)$ (1.6)

'Euclidian norm' skaler çarpım sayesinde tanımlanmakta olup, $x=(\beta_1,\beta_2,\ldots,\beta_n)$ ise,

$$norm(x)=\sqrt{\langle x,x\rangle}=\sqrt{(\beta_1)^2+(\beta_2)^2+\ldots+(\beta_n)^2} \quad (1.7)$$

şeklinde gösterilir. Dikkat edilirse Euclidian uzaklık d(x,y), Euclidian normda norm(x-y) 'ye eşit olmaktadır.

Açılar ve diklik : İki Euclidian vektör arasındaki açı ;

$$Cos\theta = \frac{\langle x,y\rangle}{norm(x)norm(y)} \quad (1.8)$$

şeklinde tanımlanır ve $\langle x,y\rangle = 0$ ise iki vektör diktir denir.

Lineer bağımsızlık : $x_1,x_2,\ldots,x_k$ vektörler olmak üzere, bu vektörlerin lineer kombinasyonu

$$\tau_1x_1+\tau_2x_2+\ldots+\tau_nx_n \quad (1.9)$$

$\tau_1=\tau_2=\ldots=\tau_n=0$ olmadıkça sıfır olamaz ise "bu vektörler lineer bağımsızdır" denir.

1.4.2. Matris Denklemleri

Pseudo inverse: Eğer σ bir skaler ise onun Pseudo inverse σ^+ şeklinde gösterilebilir ve

$$\sigma^{+} = \begin{cases} \sigma^{-1} & ,\text{eğer } \sigma \neq 0 \\ 0 & ,\text{eğer } \sigma = 0 \end{cases} \qquad (1.10)$$

Her matris için yalnız tek bir Pseudo inverse olduğu gösterilebilir.(2) Vektörler ile çalışırken örneğin x vektörünün Pseudo inverse'ı x^{+} şeklinde gösterilecektir.

$$x^{+} = \begin{cases} x^{T}/x^{T}x & ,\text{eğer } x \neq 0 \\ 0^{T} & ,x=0 \end{cases} \qquad (1.11)$$

Genel bir A matrisi için Pseudo inverse

$$A^{+} = \lim_{\Phi \to 0} (A^{T}A + \Phi^{2}I)^{-1}A^{T} \qquad (1.12)$$

$$= \lim_{\Phi \to 0} A^{T}(AA^{T} + \Phi^{2}I)^{-1} \qquad (1.13)$$

şeklindedir.

Eğer A'nın kolonları lineer olarak bağımsız ise o zaman yukarıdaki (1.12) ifadesinde $(A^{T}A)^{-1}$ mevcut olduğundan Φ yerine sıfır konulur. Eğer A'nın satırları lineer bağımsız ise (1.13) ifadesinde Φ yerine 0 konulur.

Diagonal matris $\text{diag}(\Gamma_1,\Gamma_2,\ldots,\Gamma_n)$ şeklinde gösterilecek olursa;

$$(\text{diag}(\Gamma_1,\Gamma_2,\ldots,\Gamma_n))^{+} = \text{diag}(\Gamma_1^{+},\Gamma_2^{+},\ldots,\Gamma_n^{+}) \qquad (1.14)$$

olur.

Greville Teoremi :

Genel bir matrisin Pseudo inversini hesaplamanın çeşitli yolları vardır. Programlama açısından kolaylık

getirdiginden iteratif bir algoritma olarak bilinen Greville teoremini kullanmak daha kullanışlı olacaktır.

Eger bir A matrisi k kolona sahip ise , $A_k = (A_{k-1} \quad a_k)$ şeklinde parçalanabilir. Burada A_{k-1} , k-1 kolona sahip olan bir matris ve a_k ise A_k matrisinin k ıncı kolonu olan vektör olmaktadır.

Türetme yapmaksızın A_k matrisinin Pseudo inverse'nı yazarsak;

$$A_k^+ = \frac{A_{k-1}^+(I-a_k p_k^T)}{p_k^T} \qquad (1.15)$$

olmaktadır.(2) Burada ;

$$p_k = \begin{cases} \dfrac{(I-A_{k-1}A_{k-1}^+)a_k}{\text{norm}((I-A_{k-1}A_{k-1}^+)a_k)2} & \text{, eger pay} \neq 0 \\ \dfrac{(A_{k-1}^+)^T A_{k-1}^T a_k}{1+\text{norm}(A_{k-1}^+ a_k)^2} & \text{, diger durumlarda} \end{cases} \qquad (1.16)$$

Denklem (1.15) teki iteratif eşitligin başlangıç degeri A_1, A'nın birinci kolonuna eşit olup,

$$A_1^+ = \begin{cases} a_1^T(a_1^T a_1)^{-1} & \text{, eger } a_1 \neq 0 \text{ ise} \\ 0^T & \text{, diger durumlarda} \end{cases} \qquad (1.17)$$

şeklindedir.

Pseudo inverse için bazı yararlı özellikler :

1) $0^+ = 0^T$ (0 : elemanlarının tamamı sıfır olan matris.)

2) $(A^+)^+ = A$

3) $(A^T)^+ = (A^+)^T$

4) $(\sigma A)^+ = \sigma^{-1}A^+$, eğer $\sigma \neq 0$

5) $A^+ = (A^TA)^+A^T = A^T(AA^T)^+$

6) $A^+ = A^{-1}$, eğer A kare ve tekil olmayan (non-singular) matris ise

7) $A^+ = (A^TA)^{-1}A^T$, eğer A'nın kolonları lineer bağımsız ise

8) $A^+ = A^T(AA^T)^{-1}$, eğer A'nın satırları lineer bağımsız ise

9) $r(A^+) = r(A) = r(A^T)$, r : rank

10) $A^TAA^+ = A^T$

11) $(A^+)^TA^TA = A$

12) $A^+AA^T = A^T$

13) $AA^T(A^+)^T = A$ (1.18)

1.4.3. İzdüşüm Operatörü

Keyfi bir $x \in R^n$ vektörünün $\hat{x} \in £ \subset R^n$ ve $\tilde{x}$ (dik £) dik bileşenleri lineer dönüşümler yardımıyla hesaplanabilir. Simetrik öyle bir P matrisi mevcuttur ki; $\hat{x} = Px$ $\tilde{x} = (I-P)x$ şeklinde dik bileşenler hesaplanabilir. Burada, P, £ uzayı üzerindeki ve (I-P), dik £ uzayındaki vektörle-

rin oluşturduğu izdüşüm matrisleri olmaktadır.

$x_1, x_2, \ldots, x_k$, $k < n$ kolonları olan bir X matrisini ele alalım.

$$\mathbf{x} = \hat{x} + \tilde{x} \tag{1.19}$$

$$\hat{x} = (XX^+)x \tag{1.20}$$

$$\tilde{x} = (I-XX^+)x \tag{1.21}$$

şeklinde olduğu gösterilebilir. Burada

$$P = XX^+ \quad \text{ve} \quad (I-P) = (I-XX^+) \tag{1.22}$$

olmaktadır.

Böylece modelde kullanılan matematiksel bağıntılar kısaca tanıtılmış olmaktadır. Bundan sonraki bölümlerde matematiksel model ve bu modelin programa döküIlmesi incelenecektir.

2. ÇAĞRIŞIMLI VERİ İŞLEMEDE MATEMATİKSEL MODEL

2.1. Giriş

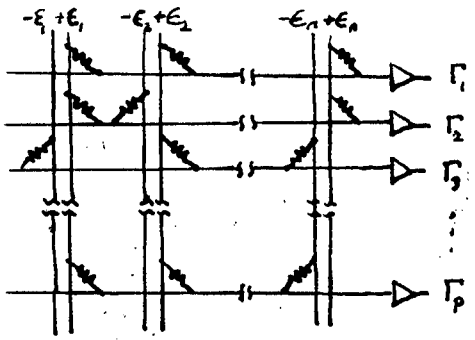
Sayısal bilgisayarlar günümüzde nümerik ve kesik değerli (discreate) hesaplamalarda çok verimli olmaktadır. Örüntü (pattern) halindeki genellikle tam tanımlanmamış veya arızalı tanımlanmış bilgiler ile çalışırken büyük veri yığınlarıyla karşı karşıyayız demektir. Böyle operasyonlar için analog metodlarla çalışan büyük paralel bilgisayarları kullanmak uygun olabilir. Ayrıca insan beyniin en önemli fonksiyonlarından biri olan hafızanın nasıl çalıştığı veya başka bir deyişle hatırlama işleminin nasıl yapıldığını incelemek sorunun çözümü konusunda fikir verebilecektir. Bunun paralelinde örüntü tanıma (pattern recognition) işlemini gerçekleştirebilmek için bilgileri işleyen, büük adaptif devrelerin kullanılması yoluyla, analog yöntem ler devreye sokulabilir. Biyolojik hafızalarda muhtemelen adaptif filtreler gibi işlemektedirler. Onların ilk amacı rassal hataları, algılama deneyimleri yardımıyla düzeltmek olarak görünmektedir.

Ö halde yapılması gereken; tanımada kullanılacak referans örüntüleri sisteme önceden bir şekilde bildirmektir. Böylece daha sonra, aynı referans örüntülerden herhangi birinin bir kısmı (veya bozuk olarak tamamı) verilerek, hafızadaki ilgili referans örüntü yardımıyla, anahtar örüntü tekrar teşkil edilebilecektir. Ayrıca tamamen hafızadaki verilerden farklı bir anahtar örüntü verilmesi halinde tanıma hafızadaki bilgiler uyarınca yapılabildiğinden iyi bir sonuç alınamayacağına dikkat etmek gerekir. Yalnız ilgili anahtar örüntü de istenirse referans bir örüntü olarak hafızaya alınabilir ve daha sonra ikinci kez gelmesi halinde tanınması mümkün olabilir. Aslında çağrışımlı hafızalarda da bir hata toleransı mevcuttur. Zira bu tip hafızalara en iyi örnek teşkil eden insan beynini ele alırsak; bazı nesneleri tanıyamama veya yanlış hatırlamak söz konusu olabilmek-

tedir. Çagrışımlı hafıza modelini gerçekleştirmekten bahsedince ilk akla gelen bir lojik devre veya operasyon kullanmak olmaktadır. Fakat bu işlem fiziksel sistemlerde basit sinyal iletişimi sayesinde de gerçekleştirilebilir.

2.2. Analog Çagrışım Modeli Olarak Direnç Devresi

Lineer bir fiziksel sistem sayesinde giriş örüntüleri çıkış örüntülerine transfer edilebilir. Örneğin eleman değerleri depolanan örüntüler yardımıyla hesaplanan bir direnç devresi böyle bir fiziksel sistemi oluşturabilir.



Şekil 2.1 Lineer dönüşüm için fiziksel model

Şekil 2.1. deki devre bir lineer dönüşümü gerçekleş - tirmektedir. Her bir yatay hattın sonundaki çıkış elemanı bir yükselteçtir ve giriş empedansı düşük op-amp gibi düşünülebilir. Çıkış voltajı düşey girişlerin bir lineer kombinasyonu olmak üzere;

$$\Gamma_i = \sum_{j=1}^{n} \sigma_{ij}\epsilon_j \qquad (2.1)$$

şeklinde yazılabilir. Buradaki σ_{ij} i'nci ve j'nci hatlar arasındaki girişlerin iletkenliği olmaktadır.

Negatif iletkenliği önlemek üzere aşağıdaki metod kullanılabilir. Her bir giriş $+\epsilon_j$ ve $-\epsilon_j$ olarak temsil edilecek olursa; katsayı σ_{ij} pozitif ise iletkenlik doğrudan $+\epsilon_j$ hattına bağlı direnç ile orantılı, σ_{ij} negatif ise iletkenlik $-\epsilon_j$ hattına bağlı direnç ile orantılı olacaktır. Bu yöntemle elde edilen akımlar çıkış yükselteçlerinde toplanmakta ve çıkış voltajlarına transfer edilmektedirler.

Burada σ_{ij} farklı ölçülere göre türetilebilir. Bu seçeneklerden biri olan en küçük kareler yöntemi optimum bir seçici olabilir.(1) Giriş vektörlerini;

$x_t=(\epsilon_{1t},\epsilon_{2t},\ldots,\epsilon_{nt})$
çıkış vektörlerini de

$Y_t=(\Gamma_{1t},\Gamma_{2t},\ldots,\Gamma_{nt})$
şeklinde gösterecek olursak,

$$\sigma_{ij} = \sum_{t=1}^{m} \Gamma_{it}\epsilon_{jt} \qquad (2.2)$$

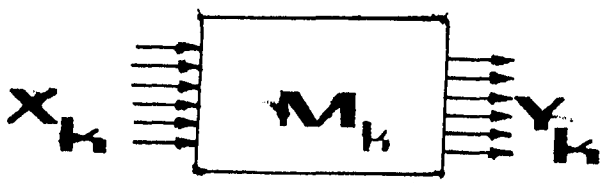
şeklinde her bir direncin değeri hesaplanabilir. Böylece verilen herhangi bir anahtar örüntü $X=(\epsilon_1,\epsilon_2,\ldots,\epsilon_n)$ girişe uygulanırsa, çıkış

$$\Gamma_i = \sum_{t=1}^{m} \left(\sum_{j=1}^{n} \epsilon_j\epsilon_{jt}\right)\Gamma_{it} = \sum_{t=1}^{m} w_t\Gamma_{it} \qquad (2.3)$$

şeklinde bulunabilecektir. Buradaki w_t katsayısı depolanmış olan örüntüler (X_t) ile anahtar örüntünün skaler çarpımına bağlı olmaktadır. Eğer anahtar örüntü depolanmış örüntülerden birine çok benziyorsa tekrar oluşturma esnasında en büyük katkıyı o örüntüden alacak ve diğer örüntülerinde az katkısıyla oluşturulacaktır.

2.3. Analog Çagrışımlı Örüntü Tanıma İçin Bir Sistem Modeli

Şekil 2.2. de görülen sistem girişteki sinyal örün - tülerini çıkışa lineer olarak transfer etmektedir.



Şekil 2.2 Çagrışımlı örüntü tanıma için bir sistem modeli.

Burada giriş sinyal örüntü vektörleri; X_k (R^n uzayında tanımlı) ve çıkış sinyalleri de; Y_k (R^P uzayında tanımlı) olarak gösterilmiştir. Örüntülerin lineer olarak transfer edildiğini düşünürsek, örüntülerin transfer ilişkisi

$$Y_k = M_k X_k \qquad (2.4)$$

şeklinde verilebilecektir. M_k matrisi pxn'lik bir matris tir. Burada araştırılacak olan M_k matrisinin var olup olmadıgıdır. Yani öyle bir M_k dönüşüm matrisi bulunabilir mi ki, verilen her giriş örüntüsüne karşın bir çıkış örüntüsü üretebilsin. Aslında M_k matrisini başlangıçta (depolanmış bilgi yok) boş kabul edip daha sonra depolanan Y_k örüntüleri yardımıyla oluşturmak söz konusudur. Bu bir iteratif problem olarak görülebilir. M_k'nın yeni optimal degeri yeni gözlenen X_k ve Y_k'nın ve M_{k-1} 'ın bir önceki optimal degerinin bir fonksiyonudur.(1) Bundan sonra fiziksel anlamdaki örüntü yerine matematiksel olarak vektör düşünülecektir ve matematiksel türetmelerde örüntünün bir vektör ile temsil edildigi kabul edilecektir.

Eğer $Y_k=M_kX_k$ matris denkleminin bir çözümü var

ise $M_k=Y_kX^+_k$ çözümü en iyi hata toleransına sahip çözüm olacaktır. Eğer tam bir çözüm yoksa $M_k=Y_kX^+_k$ en küçük kareler manasında yaklaşık en iyi çözüm olmaktadır (1).

Bir giriş matrisini (X) kabul edelim. Bunun kolonları $x_1,x_2,\ldots,x_k$ giriş vektörlerinden oluşsun. Aynı şekilde $y_1,y_2,\ldots,y_k$ vektörlerinin kolonlarını oluşturduğu bir Y çıkış matrisinin var olduğunu kabul edelim. X matrisini, X_{k-1}; k-1 kolona sahip matris ve k. ko lonun oluşturduğu bir x_k vektörü olmak üzere parçalayalım ve X_k olarak isimlendirelim. Yani $X_k = (X_{k-1} ; x_k)$ olsun. Aynı şekilde Y matrisini $Y_k = (Y_{k-1} ; y_k)$ olarak parçalayalım. Eğer M_k matrisi X_k ve Y_k dan hesaplanacak olursa; $M_k=Y_kX_k^+$ olur ve iteratif form

$$M_k = Y_kX^+_k$$

ve X^+_k yerine denk. 1.15 konulursa,

$$M_k = (Y_{k-1} ; y_k) \frac{X^+_{k-1}(I-x_kP_k^T)}{P_k^T} \qquad (2.5)$$

$$=Y_{k-1}X_{k-1}^+ + (y_k - Y_{k-1}X_{k-1}^+x_k)P_k^T$$

$$=M_{k-1} + (y_k - M_{k-1}x_k)P_k^T \qquad (2.6)$$

olur. Buradaki kazanç vektörü P_k^T denk. 1.16 'da verilen değere sahiptir. Denk. 2.6 daki iteratif denklemde iterasyon M_o= 0 'dan başlatılır. Her yeni (y_k,x_k) çifti için yeni bir M_k hesaplanmaktadır. Böylece M_k yardımıyla her yeni giriş örüntüsü (vektör) x_k için yeni bir y_k çıkış örüntüsü (vektörü) oluşturulmuş olmaktadır. Bu durum yeni x_k vektörünün ezberlenmesi olarakta görülebilir.

2.4. Ortogonal İzdüşüm

2.4.1. Ortogonal izdüşümün çağrışımlı modelde kullanımı

Ortogonal İzdüşüm : $\mathcal{L} \subset R^n$ alt uzayında m ayrı Euclidean vektörü $x_1 , x_2 , \ldots , x_m \subset R^n$ şeklinde gösterilsin. Buradaki her bir x vektörü aslında $\hat{x}$ ve $\tilde{x}$ olmak üzere ilgili vektörün dik bileşenleri cinsinden temsil edilebilir. Yani $\hat{x}$, x vektörünün $\mathcal{L}$ alt uzayındaki yatay bileşeni olmaktadır. $\tilde{x}$ ise, x vektörünün $\mathcal{L}$ alt uzayındaki dik bileşeni olmaktadır. Başka bir bakış açısından; eğer $\mathcal{L}$ alt uzayını hafızada depolanan (vektör olarak) örüntüler germekte ise; $\hat{x}$ hafızadaki örüntülere x giriş örüntüsünün benzeme miktarı, $\tilde{x}$ ise hafızadaki örüntülerde x ' in farklılık miktarı olarak değerlendirilebilir.

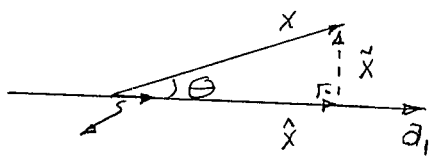
$$\hat{x} = \sum_{k=1}^{m} \Phi_k x_k \qquad (2.7)$$

şeklinde gösterecek olursak x_k depolanmış m tane ayrı (lineer bağımsız) vektördür. x vektörü de (muhtemelen tam olmayan) anahtar vektördür. Başka bir deyişle referans örüntülerini, depolanmış x_k vektörleri, x örüntüsünüde x vektörü göstermektedir. x, depolanmış x_k vektörlerinden birisine (örneğin x_r) çok benziyorsa lineer toplamdaki $\Phi_r x_r$ 'ın, $\hat{x}$ vektörüne katkısı en büyük olacaktır. Bu sayede oluşan $\hat{x}$ (yani x anahtar vektörünün x_k vektörleri yardımıyla oluşturduğu alt uzaydaki benzerlik bileşeni) vektörü x vektörüne en benzeyen vektör olup çağrışımlı hafıza modeli diyebileceğimiz yöntemle, hafızadaki tüm verilerin değerlendirilmesiyle oluşturulmaktadır.

Burada sorun ; Φ_k katsayılarının nasıl belirleneceği olmaktadır. Φ_k, Gram-Schmidth işlevi yardımıyla belirlenebilecektir.

Gram-Schmidth işlevi:

R^n'in 1 boyutlu alt uzayını alalım. Bu alt uzayın vektörü a_1 olsun ve herhangi bir x vektörünün bu uzay üzerine izdüşümünü alalım.



$$b_1 = \frac{a_1}{norm(a_1)}$$

$\tilde{x}$ = ? aranırsa $\tilde{x} = x - \hat{x}$ ifadesinden dolayı $\hat{x}$ = ? sorusuna cevap bulmamız halinde $\tilde{x}$ 'da çözülmüş olacaktır.

$$\hat{x} = kb_1 \qquad (2.8)$$

denk. 2.8 'deki k katsayısını araştırırsak;

$$\cos\theta = \frac{\langle x, \hat{x}\rangle}{norm(x)\,norm(\hat{x})} \Rightarrow \frac{norm(kb_1)}{norm(x)} = \frac{\langle x, kb_1\rangle}{norm(x)\,norm(kb_1)} \qquad (2.9)$$

$$\Rightarrow norm(kb_1)^2 = k\langle x, b_1\rangle$$

$$\Rightarrow \langle kb_1, kb_1\rangle = k\langle x, b_1\rangle$$

$$\Rightarrow k^2\underbrace{\langle b_1, b_1\rangle}_{=+1} = k\langle x, b_1\rangle$$

$$\Rightarrow k = \langle x, b_1\rangle \qquad (2.10)$$

eklinde bulunur. Denk. 2.10, denk. 2.8 'de yerine konursa,

$$\Rightarrow \hat{x} = \langle x, b_1\rangle b_1$$

ve

$$\Rightarrow \tilde{x} = x - \hat{x} = x - \langle x, b_1\rangle b_1$$

olur.

Buradan;

$$\Rightarrow x = x - \langle x, a_1 \rangle \frac{a_1}{\text{norm}(a_1)^2} \qquad (2.11)$$

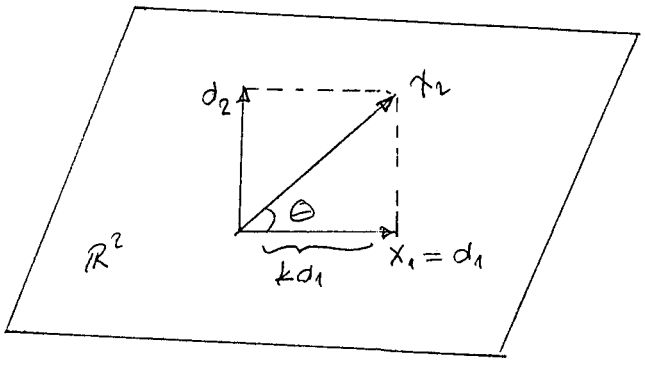
$$x = \langle x, a_1 \rangle \frac{a_1}{\text{norm}(a_1)^2} \qquad (2.12)$$

olarak 1 boyutlu alt uzayın a_1 bileşeni ve x vektörü yardımıyla verilen x vektörünün dik bileşenleri bulunur.

Herhangi bir x vektörünün £ alt uzayındaki vektörler cinsinden x ve x bileşenlerini bulmak için kullanılan £ alt uzay ortogonal olmak zorundadır. O halde şimdi R^n 1,2,... ve k+1 boyutlu alt uzayları için ortogonal birer taban bulalım.

1 boyutlu için (x_1) tabanını alalım. $x_1 = a_1$ alırsak; R^n'in iki boyutlu bir alt uzayı için (x_1, x_2) tabanını alalım. (x_1, x_2) tabanından hareketle ortogonal bir taban elde edeceğiz. Bu tabana (a_1, a_2) diyelim.

$a_1 = x_1$ alalım. a_1 vektörüne dik olan a_2 vektörü araştırılırsa;



denk.2.12 'den,

$$ka_1 = \frac{\langle x_2, a_1 \rangle a_1}{\text{norm}(a_1)^2}$$

$$a_2 = x_2 - \langle x_2, a_1 \rangle \frac{a_1}{\text{norm}(a_1)^2} \qquad (2.13)$$

olacaktır.

Bulunan bu ortogonal tabanın dikliği araştırılacak olursa;

$$\langle a_1, a_2 \rangle = \langle a_1, x_2 \rangle - \langle x_2, a_1 \rangle \frac{a_1}{\text{norm}(a_1)^2}$$

$$= \langle a_1, x_2 \rangle - \langle a_1, a_1 \rangle \frac{\langle x_2, a_1 \rangle}{\text{norm}(a_1)^2} = 0$$

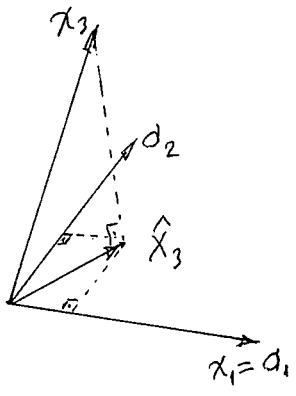
O halde a_1 ve a_2 birbirine diktir. (a_1, a_2) ortogonal bir tabandır. Şimdide R^n'in üç boyutlu bir alt uzayı için (x_1, x_2, x_3) tabanını alıp bunu ortogonalleştirelim.

$$a_1 = x_1 \qquad (2.14)$$

alalım.

$$a_2 = x_2 - \langle x_2, a_1 \rangle \frac{a_1}{\text{norm}(a_1)^2} \qquad (2.15)$$

olur ve



$$a_3 = \tilde{x}_3 = x_3 - \hat{x}_3 = x_3 - \langle x_3, a_1 \rangle \frac{a_1}{\text{norm}(a_1)^2}$$

$$- \langle x_3, a_2 \rangle \frac{a_2}{\text{norm}(a_1)^2} \qquad (2.16)$$

şeklinde bulunur.

Şimdi de bu tabanın ortogonal olup olmadığı araştırılırsa,

$$\langle a_1, a_3 \rangle = \langle a_1, x_3 \rangle - \langle x_3, a_1 \rangle \frac{a_1}{\text{norm}(a_1)^2}$$

$$- \langle x_3, a_2 \rangle \frac{a_2}{\text{norm}(a_2)^2}$$

$$= \langle a_1, x_3 \rangle - \langle a_1, a_1 \rangle \frac{\langle x_3, a_1 \rangle}{\text{norm}(a_1)^2}$$

$$- \underbrace{\langle a_1, a_2 \rangle}_{=0} \frac{\langle x_3, a_2 \rangle}{\text{norm}(a_2)^2} = 0$$

olduğundan a_1 dik a_3 tür. Aynı şekilde,

$$\langle a_2, a_3 \rangle = \langle a_2, x_3 \rangle - \langle x_3, a_1 \rangle \frac{a_1}{\text{norm}(a_1)^2}$$

$$- \langle x_3, a_2 \rangle \frac{a_2}{\text{norm}(a_2)^2}$$

$$\langle a_2,a_1\rangle = \langle a_2,x_3\rangle - \underbrace{\langle a_2,a_1\rangle}_{=0}\frac{\langle x_3,a_1\rangle}{\text{norm}(a_1)^2}$$

$$-\langle a_2,a_2\rangle\frac{\langle x_3,a_2\rangle}{\text{norm}(a_2)^2} = 0$$

a_2 dik a_3 tür. O halde, (a_1, a_2, a_3) ortogonaldir.

Şimdi bu bulduklarımızı R^n'in $(k+1)$ boyutlu alt uzayı için genelleştirebiliriz. Türetme yapmaksızın denk. 2.14, denk. 2.15, denk. 2.16 den genellemeye gidilirse;

$a_1=x_1$ ve $k=1, 2, \ldots, n-1$ için

$$a_{k+1}=x_{k+1}-\sum_{i=1}^{k}\langle x_{k+1},a_1\rangle\frac{a_i}{\text{norm}(a_i)^2} \qquad (2.17)$$

olmak üzere $(a_1,\ldots,a_{k+1})$ ortogonal bir tabandır.

Böylece bulunan x ve x vektörleri yardımıyla verilen herhangi bir x anahtar vektörünün hafızadaki vektörler yardımıyla oluşturulabilmesi söz konusu olabilmektedir.

2.4.2. Ortogonal izdüşümün hata düzeltme özelliği

Ortogonal izdüşüm metodunun, tam olmayan anahtar örüntüleri hafızadaki ezberlenmiş (referans) örüntüler yardımıyla standartlaştırdığını ve düzelttiğini söylemiştik. Bu işlem şöyle de açıklanabilir. Eğer referans örüntülerden birinin, x_r, gürültülü hali x olarak gösterilirse

$$x = x_r+v \qquad (2.18)$$

olmaktadır. Burada v stochastic hatadır. x_r'a en çok uyan, x vektörünün yatay bileşeni $\hat{x}$ olmaktadır. Basit durumda inceleme yapılacak olursa; norm(v)=v_0 olmak üzere (v sabit) alalım ve yönü de düzgün olarak R^n'de dağılmış olsun. Sonucu genellemek itibariyle v 'yi simetrik Multivariate Gaussian R^n'de radyal dağılımına uyduğunu kabul edelim(bkz.(2)). Öncelikle dikkat edilirse x_r'ın £ uzayındaki dik izdüşümü yine x_r'a eşit olacaktır. Diğer yandan v 'nin £ uzayındaki ortogonal izdüşümü $\hat{v}$, varyansı v'nin normunun karesi defa m/n olan bir dağılım gösterecektir. Burada m örüntü sayısı ,n ise onların boyutudur. Diğer bir deyişle anahtar örüntüdeki gürültü eğer m<n olursa zayıflamaktadır. Bu gürültünün standart sapması

$$Var^{1/2}(norm(x-x_r)) = \sqrt{m/n\ norm(x-x_r)} \qquad (2.19)$$

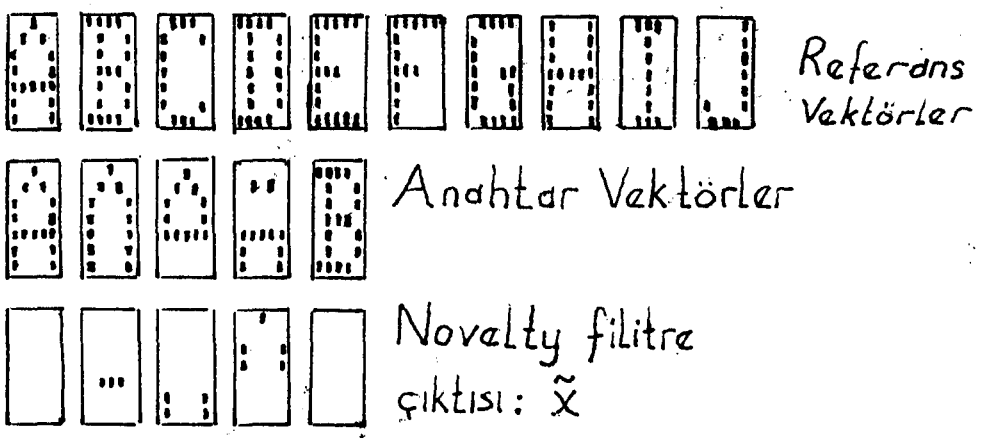
olmaktadır.

2.5. NOVELTY Süzgeci

$\tilde{x}$; x vektörünün £ altuzayındaki dik bileşeni olsun. $\tilde{x}$ 'yı,giriş verisi x'in eski örüntüler üzerindeki en iyi lineer kombinasyonundan geri kalanı olarak adlandıralım. Bu durumda $\tilde{x}$, eski vektörlerden, giriş vektörü x 'in en büyük yeniliği olarak düşünülebilecektir. Buradaki x 'in dik bileşeni ($\tilde{x}$) "Novelty" olarak adlandırılmaktadır. Novelty süzgeci ismi ise, giriş vektörü x 'den $\tilde{x}$ 'ın çıkarılması ve böylece $\tilde{x}$ bileşeni olmaksızın x vektörünü çıkışa verme işleminden gelmektedir.

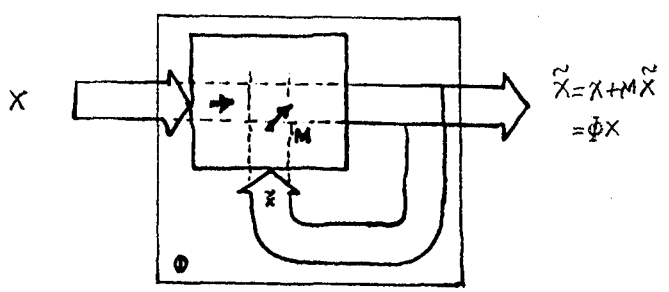
Şekil 2.3. de Novelty süzgecini daha iyi açıklamak üzere bir örnek verilmiştir(1). Örnekte on tane iki boyutlu 35 noktadan oluşan örüntü vardır. Siyah olan yerler "1" beyaz olan yerler "0" ile temsil edilmiştir. Depolanan örüntülere gösterilen girişler verildiğinde transfer matrisi yardımıyla x hesaplanmıştır.

Şekil 2.3 Novelty süzgecinin işlevi.



Şekil 2.4. deki sistem modelini ele alalım.x c R^n sistemin giriş vektörü ve $\tilde{x}$ c R^n çıkış sinyali vektörü olmaktadır. Şekilde de görüldüğü gibi çıkıştan alınan geri besleme çıkışı yine belli bir ağırlıklandırma ile etkilemektedir.

En basit durum geri beslemenin doğrudan yapıldığı halde olmaktadır. Fakat gerçekte geri besleme çıkışı sabit bir katsayı ile değil sürekli değişen ağırlıklandırma ile etkilemektedir. Bununla birlikte eğer çıkış bir alçak geçiren süzgeç üzerinden yapılırsa dinamik durumlarda da büyük değişmeler olmamaktadır. Yani geri beslemenin yine doğrudan olduğu durum haline gelmektedir.



Şekil 2.4 Novelty süzgeci için bir sistem modeli.

Çıkış vektörü $\tilde{x}$' in her bir elemanı $\tilde{\epsilon}_i$ ile gösterilecek olursa , bu her bir eleman değişebilen Γ_{ij} ağırlıklarıyla geri beslemede etkili olacaklardır. Çıkış sinyalleri $\tilde{\epsilon}_i$, giriş sinyalleri ϵ_i ve geri besleme sinyallerinin bir lineer kombinasyonu olacaktır.

$$\tilde{\epsilon}_i = \epsilon_i + \epsilon_j \Gamma_{ij} \tilde{\epsilon}_j \qquad (2.20)$$

Matris notasyonunda

$$\tilde{x} = x + M\tilde{x} = (I-M)^{-1}x = \Phi x \qquad (2.21)$$

şeklinde gösterilebilir.

Ortogonal izdüşümde, $\hat{x}$ yatay bileşenini almak ile Novelty süzgeci arasındaki asıl fark; $\tilde{x}$'ın anahtar vektörünün (x) negatif parçaları gibi üretilip, elde edilen bu vektörün anahtar vektör üzerinde bastırılmasıdır. Yani referans vektörden farklı olan kısımlar anahtar vektörden çıkarılmaktadır.

2.6. Örüntü Zenginleştirme (Pattern Enhancement)

Önceki konulara dikkat edilirse tanıma metodları özellikle anahtar vektör üzerinde rassal hatalar olduğunda optimal olmaktadır.(1) Oysa örüntü tanımanın önemli kısımlarından biri de örüntünün bir parçasından tamamını tanımaktır.

Genel halde, eğer referans örüntüler kendi içlerinde ne kadar çok dik iseler o kadar iyi bir tanıma gerçekleşmektedir. Bu gösterir ki; referans örüntüdeki keyfi bir parçanın diğer referans örüntülerin aynı parçasına mümkün olabildiği ölçüde dik olması halinde çağrışımlı yöntem daha iyi sonuç verebilecektir. Bu şekildeki örüntülerin parçaları arasındaki diklik, eğer örüntüler içinde yalnız örüntü üzerindeki hızlı değişimler, keskin hatlar varsa (yüksek yersel 'spatial' frekanslar var ise) daha fazla

olabilmektedir. Bu özelliğin sağlanması için anahtar örüntülerde olduğu kadar referans örüntülerde çağrışım işlevinden önce bir ön işleme (preprocessing) tabi tutulmalıdırlar.

Ü halde örüntü bir alçak geçiren süzgeçten geçirilmiş gibi düşünülürse; bastırılan yüksek frekansları arttırmak için bir yüksek geçiren süzgeç kullanmalıdır. Yani yüksek frekansları zenginleştiren ve böylece dikliği arttıran fonksiyonel operatörler türev ve çeşitli seviyelerdeki fark operatörleri kullanılmalıdır.

Burada görüntü zenginleştirmede kullanılan pek çok işlemden şunlar anlatılacaktır. Zira gerçek görüntüler ile (uzay foğrafları...gibi) çalışıldığında görüntü zenginleştirmeyi kullanma ihtiyacı artmaktadır.

- Komşusal işlemler

- Katlama

- Fourier dönüşümü

- Ortadeğer süzmesi

2.6.1. Komşusal işlemler

Görüntüyü netleştirmek için uygulanan yüksek geçiren süzgeç yöntemlerinde :

i. Gradient Metodunda ; i indisi ile yatay, j indisi ile düşey koordinatı tanımlanan Γ_{ij} resminin elemanı ϵ_{ij} ile gösterilirse,

$$\Gamma_{ij}=\sqrt{C^2+D^2} \qquad (2.22)$$

Burada

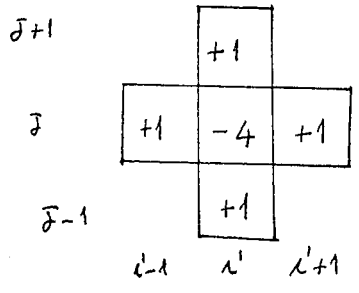
$C = (\epsilon_{i+1,j} - \epsilon_{i-1,j})/2$ ve $D = (\epsilon_{i,j+1} - \epsilon_{i,j-1})/2$

şeklinde yeniden hesaplanma ile gradient metodu uygulanmış olur.

ii. laplasyannda ise ikinci türevler bulunmakta olup

$$\sigma_{i,j} = -4\epsilon_{i,j} + \epsilon_{i,j+1} + \epsilon_{i,j-1} + \epsilon_{i+1,j} + \epsilon_{i-1,j} \quad (2.23)$$

şeklinde tanımlanır.



Şekil 2.5 Laplasyan için grid fonksiyonu.

Genelde laplasyan; bir noktadaki fonksiyon değerinin komşusal bir ortalaması ile olan farkına orantılıdır.

2.6.2. Katlama (Convolution)

Bu yöntemde görüntü işlemede kullanılan fonksiyonlardan biri orijinal görüntü, $\Gamma_{m,n}$ diğeri de süzgeç fonksiyonu f'dir. İkisinin katlanmasından süzülmüş görüntü ortaya çıkar.

$$\Gamma_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(i-m, j-n)\, \Gamma_{m,n} \quad (2.24)$$

Katlama pratikte verilen süzgeç fonksiyonunun resim üzerinde gezdirilmesinden ibaret olmaktadır.

Örnek: görüntü

$$\Gamma = \begin{bmatrix} 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \end{bmatrix}$$

süzgeç $f = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = f^t$ ise,

$$\Gamma' = \begin{bmatrix} 0 & 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 10 & 10 & 0 & 0 \\ 10 & 10 & 15 & 15 & 10 & 10 \\ 10 & 10 & 15 & 15 & 10 & 10 \\ 0 & 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 10 & 10 & 0 & 0 \end{bmatrix}$$

katlama sonucu olarak çıkar.

görüntünün boyutu yeterince büyük olduğu için görüntü boyutundaki küçülmenin pek bir zararı dokunmaz.

Çekirdek (kernel) olarakta adlandırılan süzgeç matrislerinin görüntü zenginleştirmedeki etkileri değişik çekirdeklerin kullanımıyla, kaynak 2 de belirtildiği gibi, değişik olabilmektedir.

Katlama işlemi türev almak amacıyla yapılacaksa çekirdekler; Δ^2 ikinci mertebeden türevi göstermek üzere,

Δ^2_{xf} için $[1, -1]$

Δ^2_{yf} için $\begin{bmatrix} +1 \\ -1 \end{bmatrix}$

olabilir. Katlama işlemi laplasyan (laplacian) almak için kullanılacaksa çekirdekler;

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

şeklinde olabilir. Görüntünün laplasyan kullanılarak zenginleştirilmesi halinde görüntü ile laplasyanın belli bir oranı toplanır. Bu iki işlem yukarıda belirtilen üç çekirdek için şu şekilde tek bir katlama işlemi ile gerçekleştirilebilir. Örneğin orantı katsayısı 1/2 ise,

$$\begin{bmatrix} 0 & -1/2 & 0 \\ -1/2 & 3 & -1/2 \\ 0 & -1/2 & 0 \end{bmatrix}, \begin{bmatrix} -1/2 & -1/2 & -1/2 \\ -1/2 & 5 & -1/2 \\ -1/2 & -1/2 & -1/2 \end{bmatrix}, \begin{bmatrix} -1/2 & -2 & -1/2 \\ -1/2 & 10 & -1/2 \\ -1/2 & -2 & -1/2 \end{bmatrix},$$

şeklindeki çekirdekler kullanılmalıdır.

2.6.3. Fourier Dönüşümü

Bir görüntünün Fourier dönüşümü onun içindeki yersel frekansların varlığını gösterir. İNCE(1986,S.40) ta da belirtildiği gibi Fourier dönüşümü görüntünün netleştirilmesi için kullanılabilir. " Bir görüntünün Fourier dönüşüm görüntüsü uygun bir süzgeç fonksiyonu ile çarpılıp ters dönüşüm alındığında, orijinal görüntüdeki bazı yersel frekanslar vurgulanmış bazıları da bastırılmış olarak ortaya çıkar. Görüntüde elde edilmek istenen etkiye göre uygun süzgeç görüntüleri bulmak gerekir."

2.6.4. Orta değer Süzmesi (Median filter)

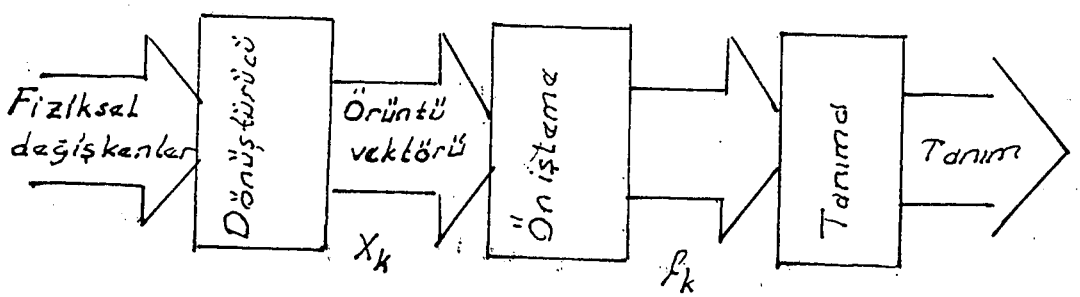
Orta değer süzmesi yardımıyla görüntüdeki nokta türü gürültüler tamamen giderilir ve aynı zamanda kenarlar korunabilir. Bu yöntemde verilen çekirdek aynen katlama işlemindeki gibi görüntü üzerinde gezdirilirken çekirdeğin altındaki değerler büyüklüklerine göre sıralanır ve ortadaki değer yeni pikselin değeri olur. Bu yöntemin dezavantajı görüntüdeki piksel kalınlığındaki çizgilerin ve keskin köşelerin kaybolmasıdır. Fakat amaca göre çekirdek değiştirilerek sakıncaların bazıları giderilebilir. Hatta İNCE(1986,S.43) tede belirtildiği gibi ;

> "Herhangi bir yakın çevre içindeki pikseller büyüklük sırasına dizilince, ortadeğer yerine başka birdeğer de alınabilir. Yakın çevrede enaza yakın bir değer almak parlak nesneleri küçültür, ençoka yakın bir değer almak ise parlak nesneleri büyültür. Karanlık bir fon üzerinde kaybolabilecek parlak bir yol en büyük değer alınarak daha görünür hale getirilebilir."

2.7. Tanıma Sisteminin Organizasyonu

Genel manada bir sistem organizasyonunu yapacak olursak; öncelikle fiziksel değişkenlerden (resim, veri ,...,vb) örüntü vektörünün elde edilmesi gereklidir. Bu örüntü vektörlerinin çağrışımlı hafıza yöntemiyle daha başarılı bir şekilde işleme alınabilmesi için vektörleri kendi aralarında daha dik hale getiren(mutual orthonormalization) veya zenginleştiren bir ön işlem gerekmektedir. Böylece ön işlemden de geçen örüntü vektörleri, daha önceden depolanmış örüntü vektörleri cinsinden tanınma yoluna gidilebilir. Bu işlemler şekil 2.4.'de görülmektedir.

Şekil 2.6 Örüntü tanıma sisteminin organizasyonu.



3. YAZILIM

3.1. Giriş

Bu bölümde daha önceki bölümlerde incelenmiş olan matematiksel modelin yazılımı akış şemaları ile anlatılmıştır. Çalışma QBASIC programlama diliyle yazılmıştır. Çalışmada kullanıcıya rahatlık getirmesi açısından mümkün olduğu ölçüde kolay ve anlaşılır (user-friendly) bir yazılım gerçekleştirilmiştir.

3.2. Ana Program

Şekil 3.1. incelenecek olursa dik izdüşüm operasyonu ve örüntü tanıma için temelde 5 ana birim mevcuttur. Bunlar ana hatlarıyla ;

SIMULATION: Referans örüntü uzayı tanımlanır.

DEFORMATION: Anahtar örüntü yaratılır.

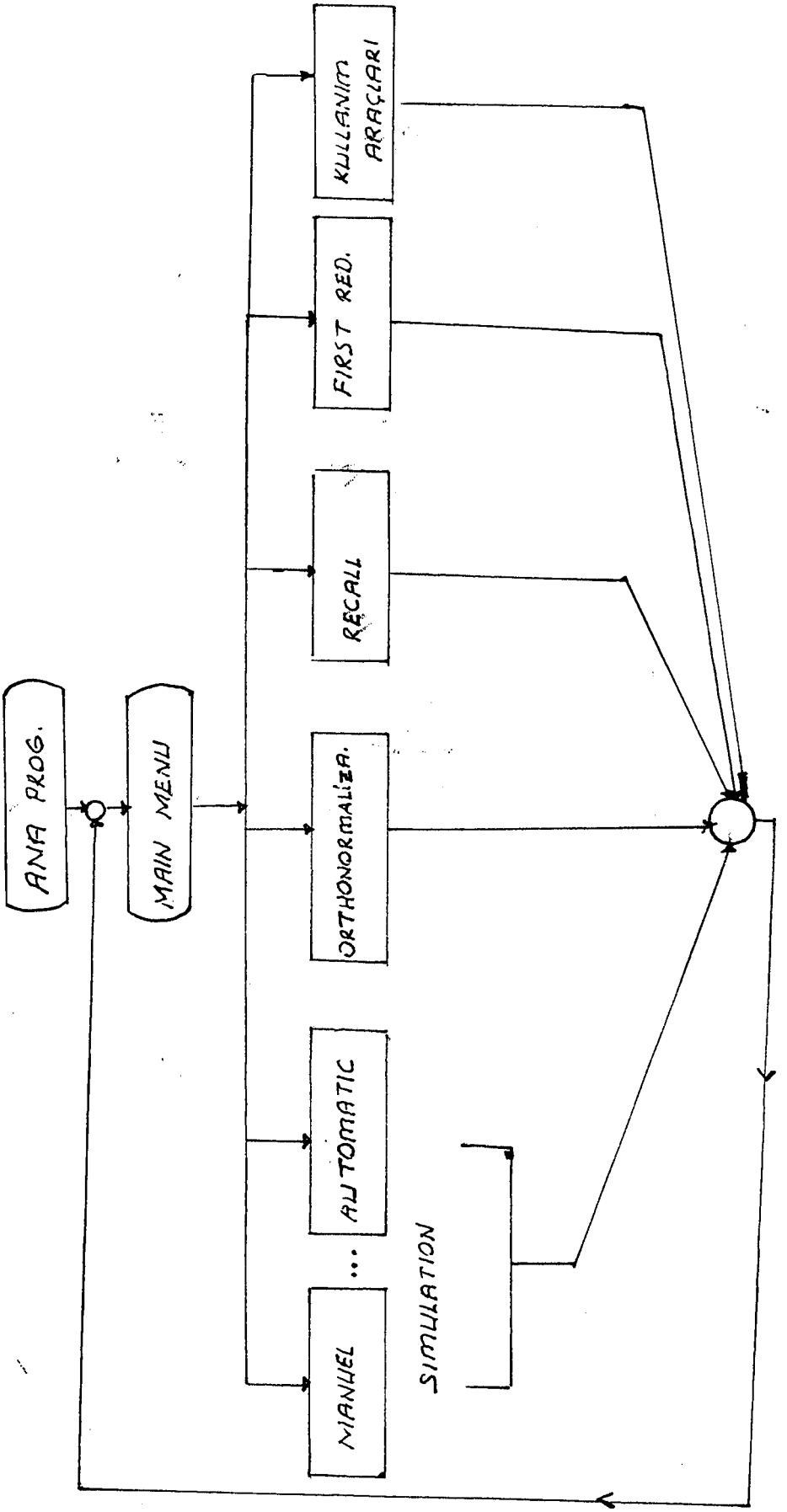
ORTHONORMALIZATION: Gramschmidth işlevi yardımıyla referans örüntü uzayı ortonormal hale getirilir.

FIRST REDUCTION (ön indirgeme): Referans örüntüler ortonormalleştirilmeden önce, kendi aralarındaki iç dikliği arttırmak üzere, ön işleme sokulurlar.

RECALL (çağrışım metodunu kullanma): Bu metod sayesinde simüle edilmiş anahtar örüntünün, önceden ezberlendiği ve hafızaya kaydedildiği varsayılan eski örüntülerden, yeniden derlenmesi işlemi yapılmaktadır.

Ayrıca kullanıcı kolaylığı sağlamak üzere kullanım araçları geliştirilmiştir.

Şimdi bu ana bloklar tek tek incelenecektir.



Sekil 3.1. Ana program akış seması

3.2.1. Simulation

Verilen bir anahtar örüntünün depolanmış örüntülerin yeni bir sentezi olarak ortaya çıkarılması manasında düşünülebilecek olan yöntemde, bilgisayar ortamında çalışıldığı için işlevin yapılabilirliğini göstermek gerekmektedir. Bu yüzden referans olarak kullanılabilecek veri kütüklerinin ve anahtar örüntünün bir şekilde bilgisayarda mevcut olması gerekmektedir. Bu veri kütükleri içinde belirli bir boyda ve sayıda elemanı olabilecek herhangi bir veri kütüğü de olabilir. Ayrıca, kolay takip edilebilirligi açısından bilgisayar ekranına bastırıldığında manalı resim teşkil eden veri kütüklerinin oluşturulması gerekmektedir. Bu şartlar göz önünde bulundurularak geliştirilen simülasyon programı yardımıyla kullanıcı isterse;

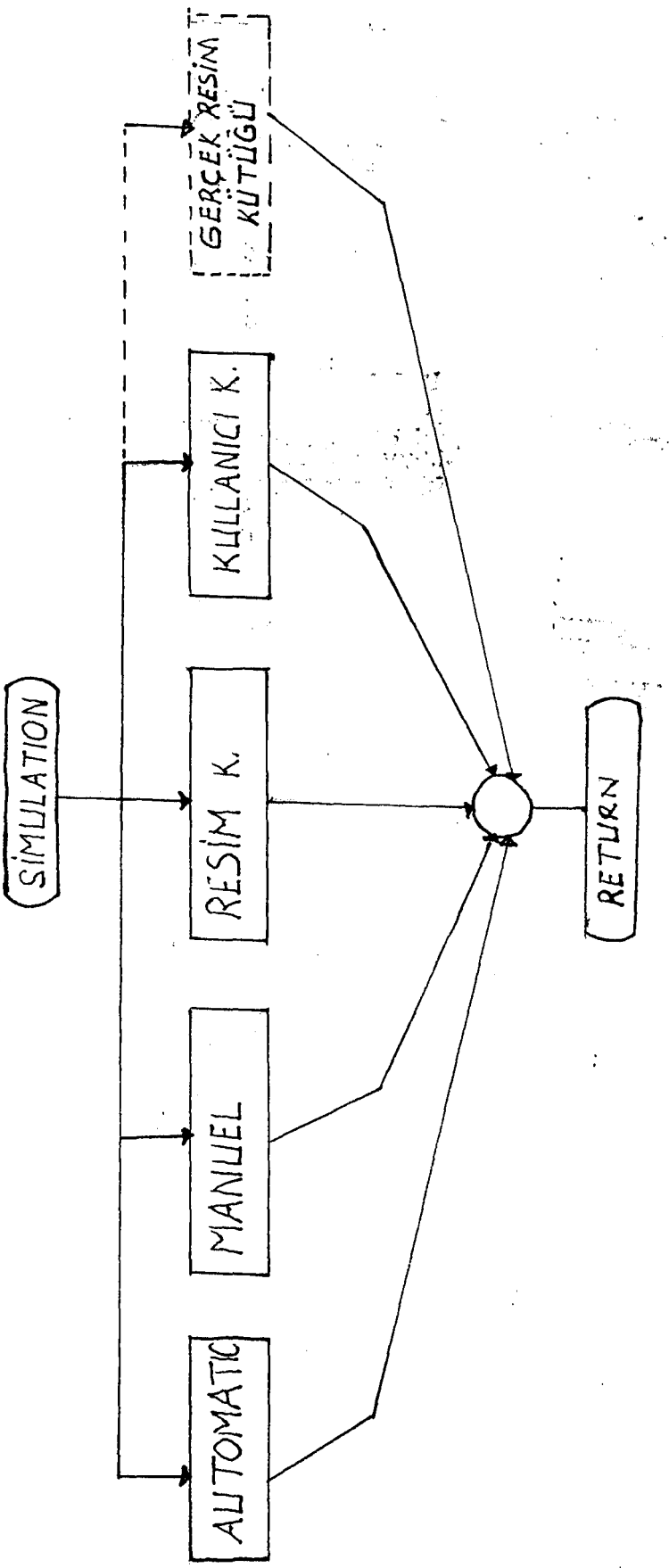
i) Vektör uzayını yaratmak üzere herbir vektörün elemanlarını kendisi bilgisayara tanımlayabilir. (MANUEL SIMULATION)

ii) Vektör uzayında kullanılabilecek vektör sayısını ve herbir vektörün boyu ile herbir elemanın hangi aralıkta olması gerektiğini belirleyerek vektör uzayını tanımlayabilir. Sonuçta verilen sınırlar dahilinde vektörler rassal olarak bilgisayarca oluşturulmaktadır. (AUTOMATIC SIMULATION)

iii) Üçüncü tip simülasyonda, bilgisayar kayıtlarında halen kayıtlı olan 120 referans örüntüden istediklerini kullanabilir ve görüntü ile çalışabilir.Bu resimler SLIB.REF ve GRAPHIC.REF kütükleri içinde bulunmaktadır.

iv) Diğer bir simulasyon türü de görüntü kütüklerini kullanıcı kendisi tanımlayabilir. (grafik editor)

Bütün bunların yanısıra eğer gerçek resimler, dijita-



Şekil 3.2. Simulation altprogramı organizasyon şeması

lize edilip bir şekilde bilgisayar ortamına sokulabilirse, çağrışımlı hafıza modeli yardımıyla işlenebilir ve verilen bir anahtar resmin hafızadaki diğer resimler arasından (varsa) bulunması istenebilir. Tabii ki işlem sonuçta referans örüntüler üzerine bir anahtar örüntünün ortogonal izdüşümünü alıp yatay bileşenini çekmek olduğuna göre, önceden yapmış olduğumuz bütün açıklamalar (gürültü, indirgeme, eksik tanımlı örüntü yardımıyla orijinal görüntüyü bulma, ...vb) bunun için de geçerlidir.
(Şekil 3.2)

3.2.2. Deformation

Deformasyon alt programı ortogonal izdüşüm almak üzere kullanılacak anahtar örüntünün yaratılması için kullanılmaktadır. Anahtar örüntü bilgisayar hafızasındaki örüntülerden herhangi birinin belirli bir oranda bozulmasıyla elde edilmektedir. Bu alt program sayesinde anahtar örüntü üzerinde istenilen oranda hata tanımlanabilir ve böylece çağrışımlı modelin kullanımından sonra hata indirgeme oranı tespit edilebilir.

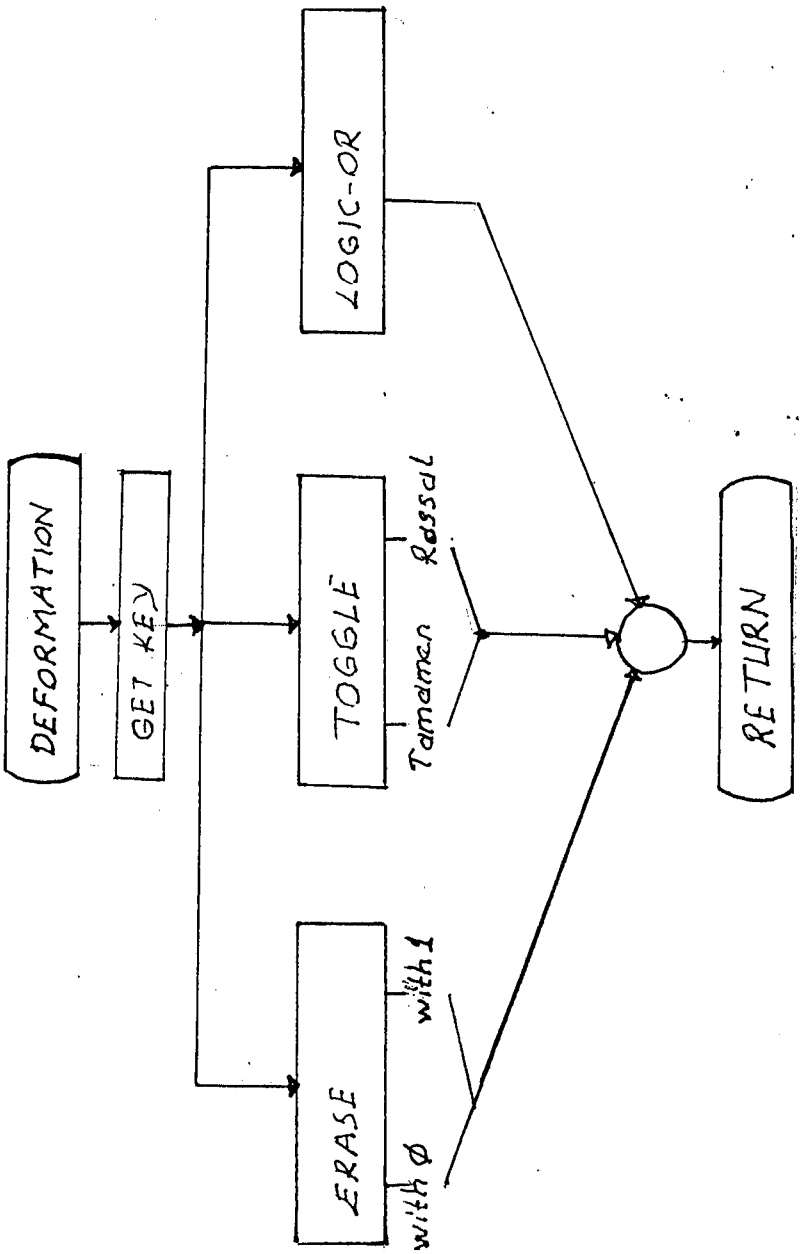
Deformasyon işlevi uygulanacak örüntü üzerinde, başlıca üç tür hata tanımlanabilir. Bunlar;

i) Örüntünün istenilen yüzdesi oranındaki kısmı,belirtilen noktadan itibaren silinir.

i.1) 0 yardımıyla: ekran görüntüsü karartılır.

i.2) 1 yardımıyla: ekran görüntüsü parlatılır.

ii) Örüntünün istenilen yüzdesi oranındaki kısmı belirtilen noktadan itibaren toggle edilir. Toggle etme eleman değerini mevcut değerinin tersi olarak alma işlemidir. Örneğin ikili sayı sisteminde çalışılıyorsa eleman değeri 1 ise 0 yapılır.



Sekil 3.3. Deformation altprogramı akış seması

ii.1) Tamamen: Verilen bölgede görüntü bilgisi toggle edilir.

ii.2) Random: Verilen bölgedeki görüntü bilgisi bilgisayarca üretilen rassal sayılara göre toggle edilir.

iii) Lojik OR'lama: Aslında görüntü bilgisinin yok edilmeyip üzerine fazladan rassal olarak hata bindirmekten ibarettir. Gerçek hata indirgemesinden ziyade verilen örüntünün nasıl derlendiğini ve ana bileşen haricindeki diğer bileşenlerin ne kadar zayıfladığını göstermesi açısından önemlidir. (şekil 3.3.)

3.2.3. ORTHONORMALIZATION

Bu alt program yardımıyla; daha önce simüle edilmiş olan referans örüntüler istenirse ön işlemden sonra ortanormalleştirilirler. (şekil 3.4.)

3.2.4. FIRST REDUCTION

Bu alt program ORTHONORMALIZATION ve RECALL tarafından kullanılmaktadır. İstenirse tanıma işlevindeki izdüşüm matrisinin daha iyi sonuç vermesi açısından referans örüntüler işlenir. Bu işlevin fonksiyonu alt bölüm 2.6 de irdelenmiştir. (şekil 3.5)

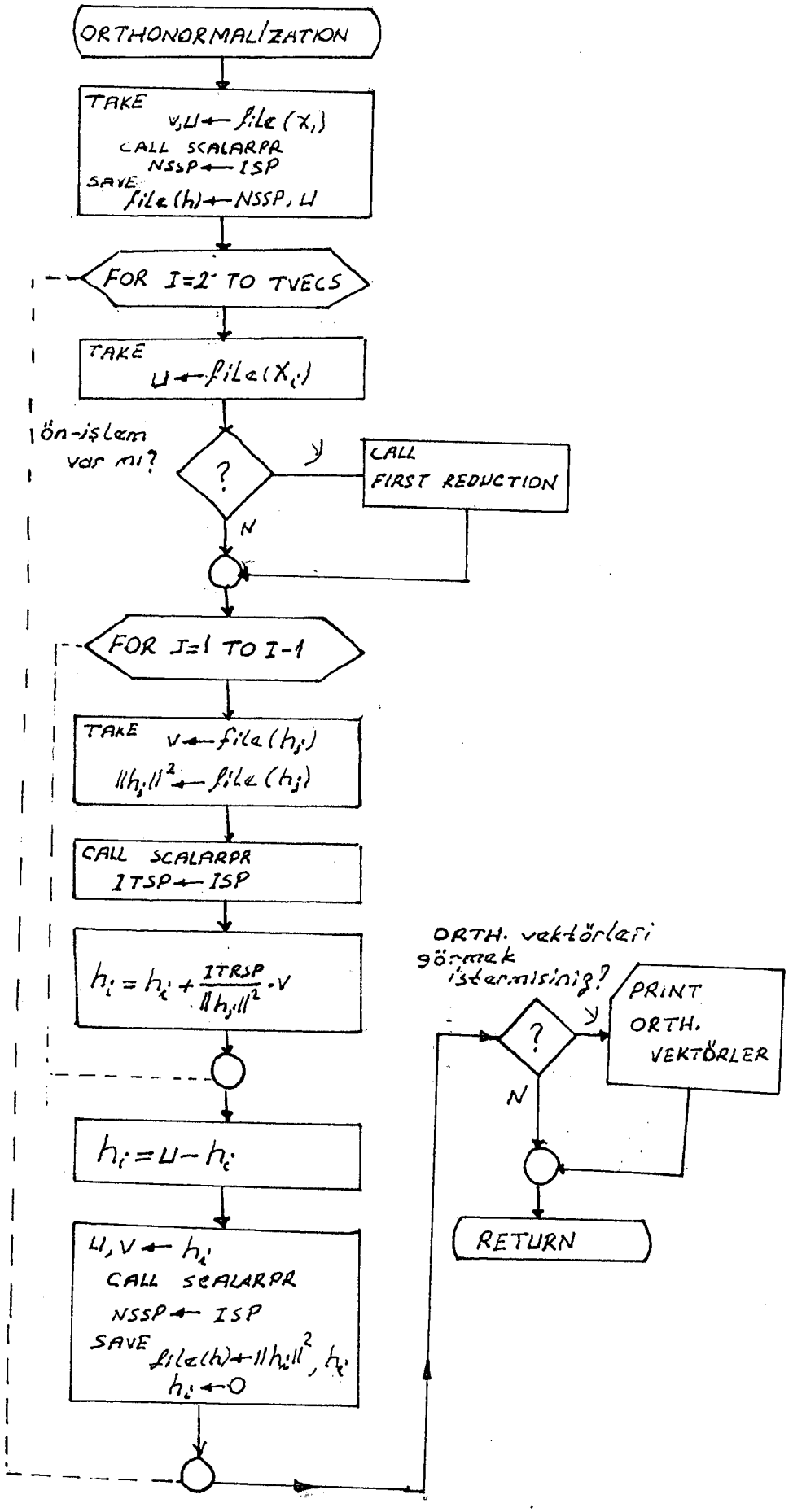
ORTHONORMALIZATION ve RECALL esnasında FIRST REDUCTION işleminde;

U(I)=f(X(G,H))

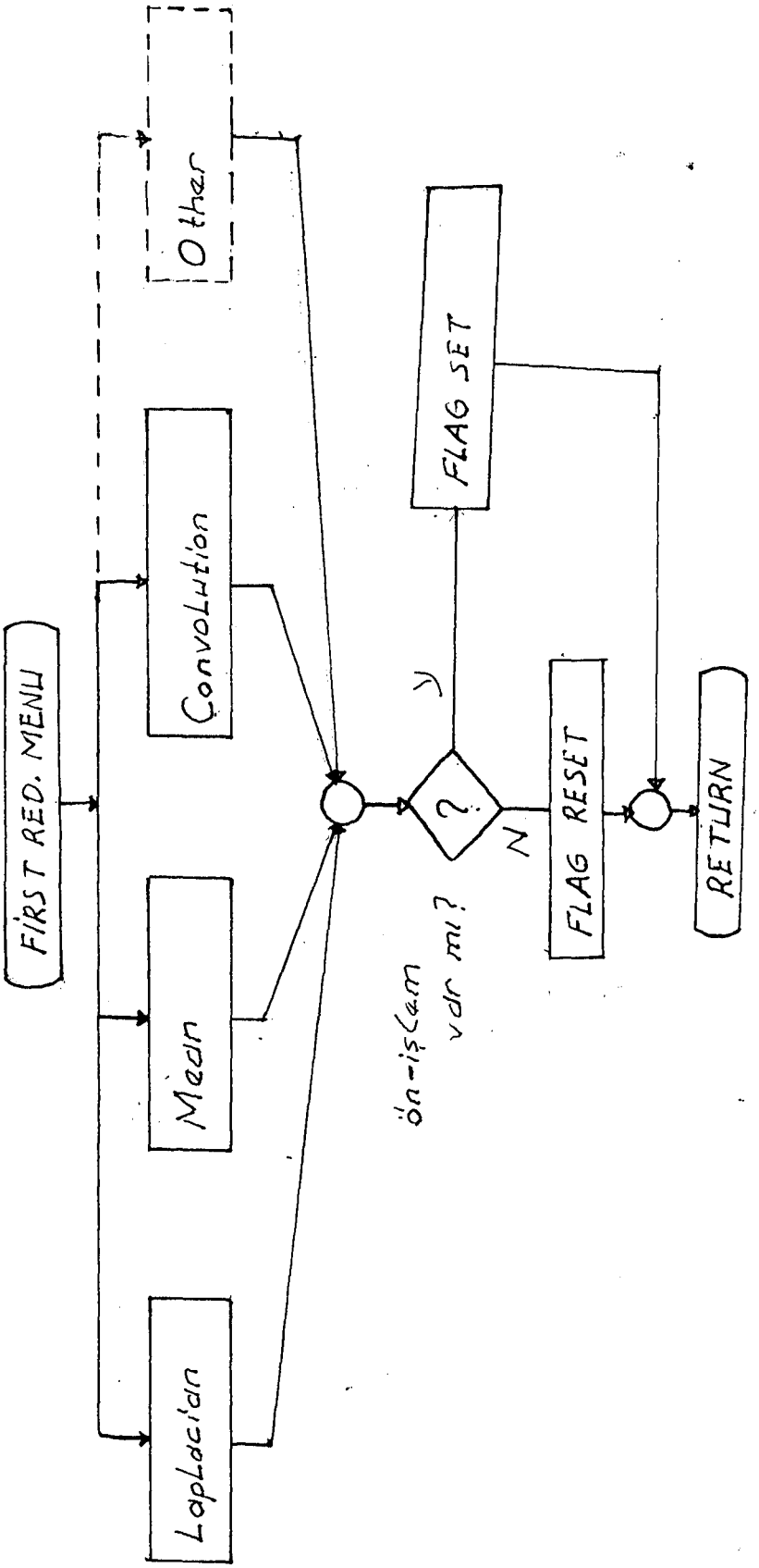
matrisi ile

OP(N X M)

matrisi kullanılmaktadır. Burada U(I) ön işleme tabi tutulacak rererans örüntüyü içinde saklar. OP(I,J) uygulanacak süzgeç fonksiyonunu içermektedir.



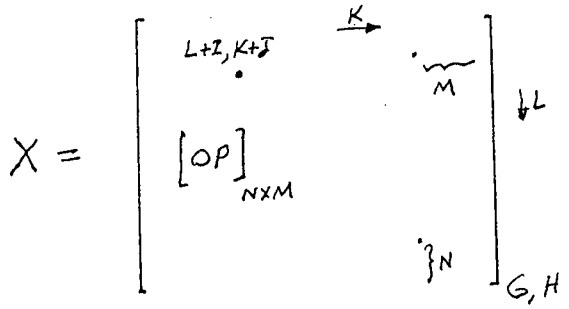
Şekil 3.4. Orthonormalization altprogramı akış şeması



Şekil 3.5. First reduction altprogramı organizasyon şeması

İşlem : X(1,1)'lik eleman ile OP(1,1)'lik eleman üst üste çakıştırılacak biçimde OP matrisi X matrisi üzerine oturtulur. Üst üste gelen elemanlar çarpılıp toplanır. Sonuç; çıktı görüntüsü, X(1,1)'in ilk değeri olur.

Sonra OP, X üzerinde bir sağa kaydırılır ve yine aynı işlemler yapılır. Böylece OP matrisi tüm X matrisi üzerinde gezdirilir. Üzerinde çalışılan örüntü matrisi aşağıdaki şekilde görüldüğü gibidir.



Üst sınır K = H-M
Alt sınır L = G-N

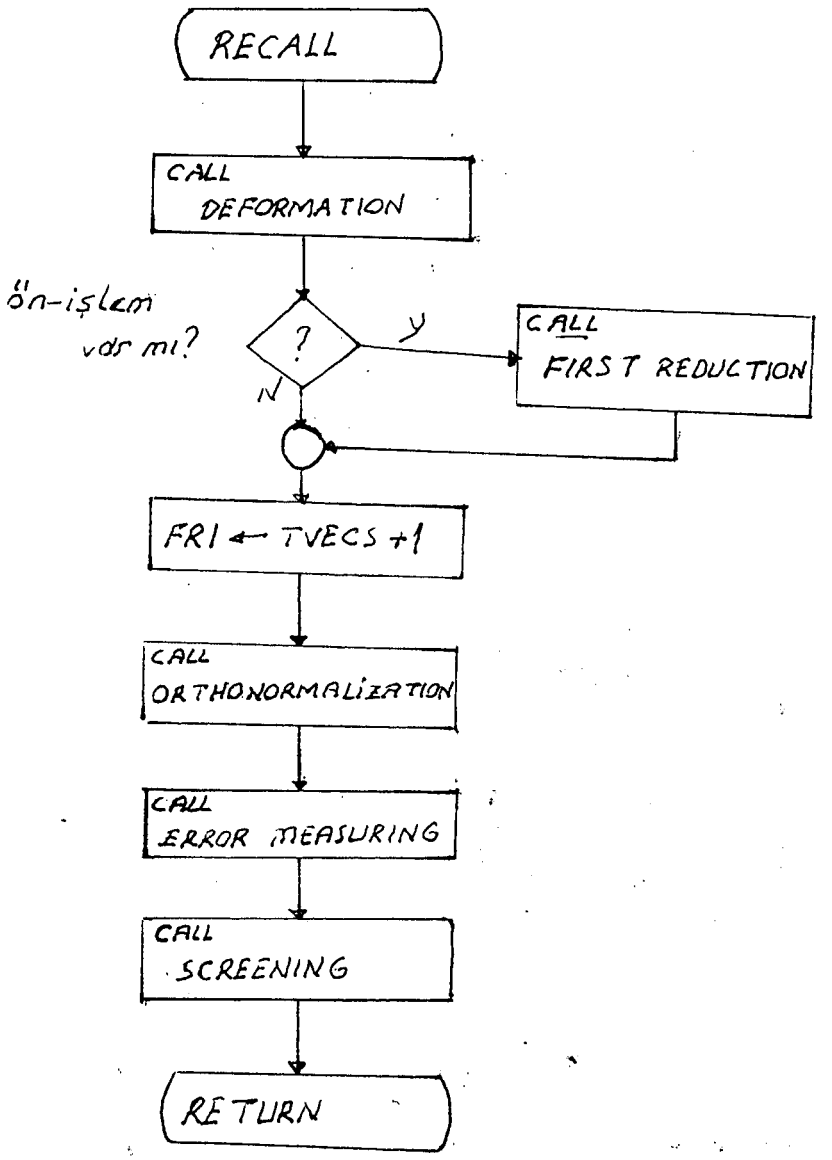
Diğer taraftan örüntü bilgileri U(K) vektöründe saklanmaktadır. Bu nedenle U(K) vektörünü X(I,J) bazında düşünmek için, örüntünün ilgili elemanını tespit etmek üzere,

$$K \equiv ((K + J) + (L + I - 1) * H) \qquad (3.1)$$

dönüşümü kullanılmıştır. Burada K; U vektörünün, I,J; X vektörünün indisleridir.

3.2.5. RECALL

Verilen bir anahtar örüntünün daha önceden depolanmış veriler cinsinden tekrar oluşturulduğu alt programdır. Burada yapılan, anahtar vektör ve hafızadaki referans örüntülerin katkısıyla yeni bir vektör oluşturmaktır. Burada sonuç vektör, ana bileşeni anahtar vektöre en çok benzeyen referans vektörden olmak üzere diğer referans vektörlerin bileşenlerinin az katkısıyla teşkil edilmektedir. Recall işleminde anahtar vektör öncelikle istenirse ön-işleme tabi tutulmakta ve daha sonradan referans örüntüler yardımıyla



Şekil 3.6. Recall altprogramı akış şeması

4. SONUÇ VE ÖNERİLER

Geliştirilen program yardımıyla bilgisayar ortamında bir vektör uzayı oluşturulabilmekte ve bu uzay Gram-schmidth işlevi yardımıyla ortogonal hale getirilebilmektedir. Daha sonra ortonormal uzay üzerine haricen tanımlanan bir vektörün izdüşümü alınabilmektedir. Tanımlanan vektör uzayı her biri bir görüntüyü içeren vektörlerden de oluşturulabilir. Ayrıca ortogonal uzay üzerine izdüşüm alınması sonucunda gürültüyü indirgeme yüzdesi de tespit edilebilmektedir.

ORTHONORMALIZATION ve RECALL altprogramlarının birlikte kullanımı sayesinde; herbiri bir vektör olarak görülebilecek pekçok veri arasından haricen tanımlanabilecek (bilgisayar ortamında interaktif tanımlama için gerekli alt programlar mevcuttur) anahtar veri aranabilir veya başka bir deyişle hafızadaki veriler yardımıyla anahtar veri (hafızada var ise) gürültüsü indirgenmiş bir şekilde yeniden oluşturulabilir.

Ayrıca programın değişik kısımları farklı amaçlarla kullanılmak üzere servis altprogramları olarak da kullanılabilir.

- Bir matematikçinin SIMULATION altprogramı yardımıyla bir vektör uzayı tanımlaması ve ORTHONORMALIZATION yardımıyla da tanımladığı vektör uzayını ortogonal hale getirmesi mümkündür.

- Görüntü zenginleştirme (FIRST REDUCTION) altprogramı ile de ortak-diklik (mutual ortogonalization) arttırılabilir. Bu yöntem örüntü sınıflama (pattern classification) çalışmalarında servis altprogramı rolünü oynayabilir.

Geliştirilen programda bilhassa görüntü işleme üzerine çalışılmış olmakla birlikte programda herhangi bir değişikliğe gerek duyulmadan farklı veri yığınlarıyla da çalışı

labilir.

Beyin araştırmaları paralelinde ortaya çıkmış olan çalışma; beyin fonksiyonlarından en önemlisi olan hafıza biriminin matematiksel modelinin ne olabileceği ve pratikte nasıl gerçekleştirilebileceği konusunda atılan küçük ama uygulamalı bir adım olarak da görülebilir. Ancak çalışma bu alanda tam amacına erişebilmiş sayılamaz. Zira hafızanın unutma mekanizmasının da model içinde incelenmesi gerekmektedir. Bu çalışmanın devamı olabilecek araştırmalarda daha genel hafıza modellemeleri gerçekleştirilebilir. Yalnız bu tip çalışmalarda beynin hangi girdilere hangi tepkileri vermekte (fiziksel davranışlar veya elektronik olarak algılanabilecek sinyaller) olduğu konusunda detaylı bir analize ihtiyaç vardır.

Ayrıca PC ile uyumlu olarak görüntünün dijitalize edilerek hafızaya atılabilmesi halinde değişik alanlarda kullanılabilecek görüntüyü gürültüden arındırma ve görüntüyü hafızada depolu diğer veriler arasından tanıma şeklinde pratik uygulamalar da yapılabilecektir.

Bu noktada gördüğümüz gürültü indirgeme oranının m/n ile orantılı olması (alt bölüm 2.4.2.) programın bir dezavantajı olarak karşımıza çıkmaktadır. Burada, m örüntü sayısı, n herbir örüntünün boyu olmaktadır. Fakat daha büyük hafıza kapasiteli bilgisayarlar yardımıyla çalışılması halinde, n büyük tutularak bu dezavantaj biraz daha azaltılabilecektir.

Ayrıca bu çalışmanın devamında geliştirilecek başka bir çalışma da programın tamamının assembler diline dökülmesi olabilir. Program her ne kadar **QBASIC** gibi yüksek performanslı (hızlı) bir dil ile geliştirilmişse de assemblerde yazılabilecek programlar daha hızlı olabilecektir.

KAYNAKLAR DİZİNİ

1. Fu,K.S., Keidel, W.D.,Levelt,W.J.M and Wolter, H.,1978,Associative memory,A system-theoretical approach,Springer-Verlag Berlin Heidelberg Newyork,176 s.

2. İnce,F.,1986,Uzaktan algılamada sayısal görüntü işleme, TÜBİTAK.

3. Lang,S., 1973,Linear algebra,Addison-Wesley Publishing Company,394 S.

4. İnce,F.,1979,Ana bileşenler,sınıflandırma ve kümeleme teknikleri,TÜBİTAK.

5. 1982,Basic reference,IBM Corp., Personal Computer , Florida,374 s.

6. Andrews.H.C.,1972, Introduction to mathematical techniques in pattern recognation, Wiley-Interscience,242.S.

EK-A PROGRAM LİSTESİ

```
10 'This Program is created by:N.ADAR
20 'PROGRAM NAME:NORTOS3.BAS
30 'VARIABLES    :-TVECS:Toplam vektor sayisi.
40 '              -XVECB:Vektorlerin boyutu.
50 '              -NSSP :Norm Square Scalar Product.
60 '              -ISP  :Immediate Scalar Product.
70 '              -ITRSP:Iterative Scalar Product.
80 '              -FR???:FOR-NEXT degiskeni.
90 '              -NSH  :Norm Square Orthonormal vector.
100 'ARRAYS        :-TH(XVECB)  :Temprory Ortonormal Vector.
110 '             :-TX(XVECB)  :Temprory Reference pattern vektoru.
120 '             :-V,U        :Scalar Product vectors.
130 '             :-IMIN,TMIN  :Gecis vektorleri.
140 '             :-H(XVECB)   :Ortonormal vektor.
150 'FILES        :XVECS.DAT Ç HVECS.DAT
160 '             -random files.
170 '             -variables:FX$,FH$
180 '             -record lengths:4
190 '             -icerik:Single precision numeric values.
200 FRFLAG$="OFF"
210 BC=4
220 FC=7'COLOR SETTING
230 REF1$="SLIB.REF"
240 REF2$="HPVECS1.ORT"
250 REFS1$="XVECS.REF"
260 TVECS=3
270 XVECB=3520
280 ITVECS=3        'orthonormal uzayin boyut ayarlamasi.
290 CLS
300 FLAG=0
310 ATRECAL=0
320 KEY OFF
330 DIM U(3520),V(3520),H(3520),OP(3,3)
340 'ÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉ SIMULATION MENU ÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉ
350 SCREEN 0
360 COLOR FC,BC
370 CLS
380 OPEN "I",#1,"ORTOS.OVL"
390 FOR I=1 TO 10
400 LINE INPUT#1,OV$
410 LOCATE 2*I-1,25
420 PRINT OV$
430 NEXT I
440 CLOSE #1
450 LOCATE 21,20
460 PRINT" SELECT ONE :";'YK
470 SEL$=""
480 WHILE SEL$=""
490 SEL$=INKEY$
500 WEND
```

```
510 YK=VAL(SEL$)
520 PRINT SEL$
530 IF YK<1 OR YK>9 THEN LOCATE 1,1:PRINT"E R R O R  !!!":BEEP:GOTO 380
540 CLS
550 YK=INT(YK)
560 ON YK GOSUB 2540,2900,3310,4130,600,1220,1440,1640,2530
570 GOTO 340
580 YK=0
590 GOTO 530
600 '********** FILE ******************************
610 OPEN"I",#1,"ORTOS.OVL"
620 FOR I=1 TO 19
630 LINE INPUT#1,OV$
640 NEXT I
650 FOR I=1 TO 6
660 LINE INPUT#1,OV$
670 LOCATE 2*I,25
680 PRINT OV$
690 NEXT I
700 CLOSE #1
710 LOCATE 15,20
720 PRINT" SELECT ONE :";'YK
730 SEL$=""
740 WHILE SEL$=""
750 SEL$=INKEY$
760 WEND
770 YK=VAL(SEL$)
780 PRINT SEL$
790 IF YK<1 OR YK>5 THEN LOCATE 1,1:PRINT"E R R O R  !!!":BEEP:GOTO 710
800 ON YK GOTO 820,840,900,2460,810
810 RETURN
820 FILES "*.REF"
830 GOTO 850
840 FILES "*.ORT"
850 PRINT"PRESS ANY KEY WHEN READY"
860 WHILE INKEY$=""
870 WEND
880 CLS
890 GOTO 600
900 LOCATE 15,10
910 PRINT"REFERENCE FILE NAME default <";REF1$;">";
920 INPUT"",SEL$
930 IF SEL$<>"" THEN REF1$=SEL$
940 LOCATE 16,10
950 PRINT"ORTHONORMAL FILE NAME default <";REF2$;">";
960 INPUT"",SEL$
970 IF SEL$<>"" THEN REF2$=SEL$
980 LOCATE 17,10
990 PRINT"SIMULATION REF. FILE NAME default <";REFS1$;">";
1000 INPUT"",SEL$
```

```
1010 IF SEL$<>"" THEN REFS1$=SEL$
1020 LOCATE 19,10
1030 PRINT"NUMBER OF TOTAL SIMULATION PATTERN default <";TVECS;">";
1040 INPUT"",SEL$
1050 IF SEL$<>"" THEN TVECS=VAL(SEL$)
1060 IF TVECS<1 THEN 1020
1070 LOCATE 20,10
1080 PRINT"LENGHT OF EACH  SIMULATION PATTERN default <";XVECB;">";
1090 INPUT"",SEL$
1100 IF SEL$<>"" THEN XVECB=VAL(SEL$)
1110 IF XVECB<1 THEN 1070
1120 LOCATE 22,10
1130 PRINT"ACTIVE ORTHONORMAL VECTOR NUMBER IN FILE ";
1140 PRINT"default<";ITVECS;">";
1150 INPUT"",SEL$
1160 IF SEL$<>"" THEN ITVECS=VAL(SEL$):GOSUB 7550
1170 IF ITVECS<1 THEN 1120
1180 FOR I=1 TO 1000
1190 NEXT I
1200 CLS
1210 GOTO 600
1220 'COLOR
1230 OPEN "I",#1,"ORTOS.OVL"
1240 FOR I=1 TO 29
1250 LINE INPUT#1,OV$
1260 NEXT I
1270 FOR I=1 TO 9
1280 LOCATE 2+I,10
1290 LINE INPUT#1,OV$
1300 PRINT OV$
1310 NEXT I
1320 CLOSE #1
1330 LOCATE 15,8
1340 PRINT "FOREGROUND COLOR default <";FC;">";
1350 INPUT"",SEL$
1360 IF SEL$<>"" THEN FC=VAL(SEL$)
1370 IF FC<0 OR FC>15 THEN BEEP:GOTO 1330
1380 LOCATE 17,8
1390 PRINT "BACKGROUND COLOR default <";BC;">";
1400 INPUT"",SEL$
1410 IF SEL$<>"" THEN BC=VAL(SEL$)
1420 IF BC<0 OR BC>15 THEN BEEP:GOTO 1380
1430 RETURN
1440 'HELP
1450 OPEN "I",#1,"ORTOS.HLP"
1460 CLS
1470 FOR I=1 TO 20
1480 LINE INPUT#1,OV$
1490 IF EOF(1) THEN 1580
1500 LOCATE I,1
```

```
1510 PRINT OV$
1520 NEXT I
1530 PRINT
1540 PRINT"PRESS ANY KEY TO CONTINUE"
1550 WHILE INKEY$=""
1560 WEND
1570 GOTO 1460
1580 PRINT
1590 PRINT"PRESS ANY KEY RETURN TO MAIN MENU"
1600 WHILE INKEY$=""
1610 WEND
1620 CLOSE #1
1630 RETURN
1640 'FIRST RED.
1650 DATA 0,-.5,0,-.5,3,-.5,0,-.5,0,1,1,1,1,1,1,1,1,1
1660 RESTORE 1650
1670 LOCATE 1,20
1680 PRINT "LAPLACIAN"
1690 FOR I=1 TO 3
1700 FOR J=1 TO 3
1710 LOCATE 1+2*I,14+5*J
1720 READ A
1730 PRINT A
1740 NEXT J
1750 NEXT I
1760 LOCATE 1,40
1770 PRINT "CONVOLUTION"
1780 FOR I=1 TO 3
1790 FOR J=1 TO 3
1800 LOCATE 1+2*I,34+5*J
1810 READ A
1820 PRINT A
1830 NEXT J
1840 NEXT I
1850 LOCATE 1,60
1860 PRINT "USER DEF"
1870 FOR I=1 TO 3
1880 FOR J=1 TO 3
1890 LOCATE 1+2*I,54+5*J
1900 PRINT "-"
1910 NEXT J
1920 NEXT I
1930 LOCATE 12,10
1940 PRINT"1.LAPLACIAN"
1950 LOCATE 14,10
1960 PRINT"2.MEAN"
1970 LOCATE 16,10
1980 PRINT"3.CONVOLUTION"
1990 LOCATE 18,10
2000 PRINT"4.USER DEFINED CORE"
```

```
2010 LOCATE 20,10
2020 PRINT"5.FIRST REDUCTION FLAG OFF"
2030 LOCATE 22,10
2040 PRINT"SELECT ONE"
2050 SEL$=""
2060 WHILE SEL$=""
2070 SEL$=INKEY$
2080 WEND
2090 IF SEL$<"1" OR SEL$>"5" THEN 2050
2100 FR$=SEL$
2110 ON VAL(SEL$) GOTO 2120,2230,2250,2360,2440
2120 FRFLAG$="ON"
2130 OP(1,1)=0
2140 OP(1,2)=-.5
2150 OP(1,3)=0
2160 OP(2,1)=-.5
2170 OP(2,2)=3
2180 OP(2,3)=-.5
2190 OP(3,1)=0
2200 OP(3,2)=-.5
2210 OP(3,3)=0
2220 RETURN
2230 FRFLAG$="ON"
2240 RETURN
2250 FRFLAG$="ON"
2260 OP(1,1)=1
2270 OP(1,2)=1
2280 OP(1,3)=1
2290 OP(2,1)=1
2300 OP(2,2)=1
2310 OP(2,3)=1
2320 OP(3,1)=1
2330 OP(3,2)=1
2340 OP(3,3)=1
2350 RETURN
2360 FRFLAG$="ON"
2370 FOR I=1 TO 3
2380 FOR J=1 TO 3
2390 LOCATE 1+2*I,54+5*J
2400 INPUT"",OP(I,J)
2410 NEXT J
2420 NEXT I
2430 RETURN
2440 FRFLAG$="OFF"
2450 RETURN
2460 OPEN "O",#1,"C:¥PMASTER¥GEDIT.SHP"
2470 FOR I=1 TO 577
2480 A$=CHR$(0)
2490 PRINT #1,A$
2500 NEXT I
```

```
2510 CLOSE #1
2520 END
2530 END'##########################################
2540 '********** SUB MANUEL SIMULATION. **********
2550 OPEN "R",#1,REF1$,4
2560 FIELD #1,4 AS FX$
2570 LOCATE 13,10
2580 PRINT "REFERENCE FILE NAME      :";REF1$
2590 LOCATE 15,10
2600 PRINT "NUMBER OF TOTAL PATTERN :";TVECS
2610 LOCATE 17,10
2620 PRINT "LENGTH OF EACH  PATTERN :";XVECB
2630 LOCATE 20,10
2640 PRINT "IS IT CORRECT ?  (Y/N)  :";
2650 SEL$=""
2660 WHILE SEL$=""
2670 SEL$=INKEY$
2680 WEND
2690 PRINT SEL$
2700 IF SEL$="Y" OR SEL$="y" THEN 2710 ELSE CLOSE#1:RETURN
2710 LSET FX$=MKS$(TVECS)
2720 PUT #1,1
2730 LSET FX$=MKS$(XVECB)
2740 PUT #1,2
2750 J=1
2760 FOR FRT=J TO TVECS
2770 CLS
2780 ARAAT4=3+(FRT-1)*XVECB
2790 I=1
2800 FOR FRX=ARAAT4 TO XVECB+ARAAT4-1
2810 PRINT FRT;".VECTOR,";I;".ELEMENT:";
2820 INPUT VERI
2830 LSET FX$=MKS$(VERI)
2840 PUT #1,FRX
2850 I=I+1
2860 NEXT FRX
2870 NEXT FRT
2880 CLOSE #1
2890 RETURN
2900 '************ SUB AUTOMATIC SIMULATION. ***************
2910 OPEN "R",#1,REF1$,4
2920 FIELD #1,4 AS FX$
2930 LOCATE 13,10
2940 PRINT "REFERENCE FILE NAME      :";REF1$
2950 LOCATE 15,10
2960 PRINT "NUMBER OF TOTAL PATTERN :";TVECS
2970 LOCATE 17,10
2980 PRINT "LENGHT OF EACH  PATTERN :";XVECB
2990 LOCATE 20,10
3000 PRINT "IS IT CORRECT ?  (Y/N)  :";
```

```
3010 SEL$=""
3020 WHILE SEL$=""
3030 SEL$=INKEY$
3040 WEND
3050 PRINT SEL$
3060 IF SEL$="Y" OR SEL$="y" THEN 3070 ELSE CLOSE#1:RETURN
3070 LSET FX$=MKS$(TVECS)
3080 PUT #1,1
3090 LSET FX$=MKS$(XVECB)
3100 PUT #1,2
3110 LOCATE 22,10
3120 INPUT "RANGE OF EACH ELEMENT (0-??)",RGR
3130 CLS
3140 LOCATE 13,35
3150 PRINT"PLEASE WAIT."
3160 FOR FRT=1 TO TVECS
3170 LOCATE 2,1
3180 PRINT FRT
3190 ARAAT5=3+(FRT-1)*XVECB
3200 I=1
3210 RANDOMIZE TIMER
3220 FOR FRX=ARAAT5 TO XVECB+ARAAT5-1
3230 VERI=INT(RND*(RGR+1))
3240 LSET FX$=MKS$(VERI)
3250 PUT #1,FRX
3260 I=I+1
3270 NEXT FRX
3280 NEXT FRT
3290 CLOSE #1
3300 RETURN
3310 '**************** SUB ORTONORMALLESTIRME ***************
3320 ATRECAL=0
3330 LOCATE 10,15
3340 PRINT"REFERENCE PATTERN FILE NAME:";REF1$
3350 LOCATE 12,15
3360 PRINT"       ORTHONORMAL FILE NAME:";REF2$
3370 PRINT
3380 PRINT
3390 PRINT "              IS IT CORRECT ?  (Y/N) :";
3400 YN$=INKEY$
3410 IF YN$="Y" OR YN$="y" THEN 3430
3420 IF YN$="N" OR YN$="n" THEN RETURN ELSE 3400
3430 CLS
3440 LOCATE 17,1
3450 PRINT"ORTHONORMALIZATION :"
3460 LOCATE 17,22
3470 PRINT"1. in USER FILE"
3480 LOCATE 19,22
3490 PRINT"2. in PMASTER FILE"
3500 PRINT
```

```
3510 PRINT"                              SELECT ONE :";
3520 SEL$=""
3530 WHILE SEL$=""
3540 SEL$=INKEY$
3550 WEND
3560 XYK=VAL(SEL$)
3570 IF XYK<1 OR XYK>2 THEN LOCATE 1,1:BEEP:PRINT"E R R O R  !!!":GOTO 34
3580 CLS
3590 ORTH=XYK
3600 PVECS=ITVECS
3610 PVECB=88*40
3620 '£££££££££££ ORTOGONALLESTIRME OPERASYONU £££££££££££
3630 ON ORTH GOTO 3640,3710
3640 OPEN "R",#1,REF1$,4
3650 FIELD #1,4 AS FX$
3660 GET #1,1
3670 TVECS=CVS(FX$)
3680 GET #1,2
3690 XVECB=CVS(FX$)
3700 GOTO 3750
3710 OPEN "R",#1,REF1$,1
3720 FIELD #1,1 AS FP$
3730 TVECS=PVECS
3740 XVECB=PVECB
3750 IF ATRECAL=1 THEN RETURN
3760 OPEN "R",#2,REF2$,4
3770 FIELD #2,4 AS FH$
3780 LSET FH$=MKS$(TVECS)
3790 PUT #2,1
3800 LSET FH$=MKS$(XVECB)
3810 PUT #2,2
3820 '#####################################
3830 FOR FRI=1 TO TVECS
3840 LOCATE 1,1
3850 PRINT FRI
3860 ON ORTH GOTO 3870,3890
3870 GOSUB 5560                 'X CEK
3880 GOTO 3900
3890 GOSUB 5970                 'PX CEK.
3900 GOSUB 5660                 ' SUB orth(X) hesapla.
3910 ARAAT3=(FRI-1)*XVECB+3+FRI
3920 LSET FH$=MKS$(NSSP)
3930 PUT #2,ARAAT3-1
3940 I=1
3950 FOR FRS=ARAAT3 TO XVECB+ARAAT3-1
3960 LSET FH$=MKS$(H(I))
3970 PUT #2,FRS
3980 I=I+1
3990 NEXT FRS
4000 NEXT FRI
```

```
4010 CLS
4020 LOCATE 10,10
4030 PRINT"DO YOU WANT SEE ORTHONORMAL VECTORS ?(y/n)"
4040 YN$=INKEY$
4050 IF YN$="Y" OR YN$="y" THEN 4070
4060 IF YN$="N" OR YN$="n" THEN 4110 ELSE 4040
4070 GOSUB 5280      'sub orth. vec. meters.
4080 PRINT
4090 PRINT"PRESS SPACE BAR TO CONTINUE."
4100 IF INKEY$=" " THEN 4110 ELSE 4100
4110 CLOSE #1,#2
4120 RETURN
4130 '*************** sub recall ******************************
4140 LOCATE 11,37
4150 PRINT"RECALL"
4160 LOCATE 17,1
4170 PRINT"RECALL :"
4180 ATRECAL=1
4190 GOSUB 3460          'FILE OPEN FROM ORTH.
4200 OPEN "R",#3,"HBUF.BUF",4
4210 FIELD #3,4 AS FHP$
4220 OPEN "R",#2,REF2$,4
4230 FIELD #2,4 AS FH$
4240 GET #2,1
4250 TVECS=CVS(FH$)
4260 GET #2,2
4270 XVECB=CVS(FH$)
4280 CLS
4290 LOCATE 15,10
4300 INPUT"KEY VECTOR NUMBER :",KVNUM
4310 FRI=KVNUM
4320 ON ORTH GOTO 4330,4350
4330 GOSUB 5560           'X CEK.
4340 GOTO 4430
4350 GOSUB 5970           'PX CEK.
4360 SCREEN 2
4370 CLS
4380 OFSY=0
4390 OFSX=0
4400 LOCATE 12,5
4410 PRINT "ORIGINAL PICTURE"
4420 GOSUB 6180'ORIGINAL KEY SCREENING.
4430 GOSUB 6650
4440 GOSUB 4850
4450 VIEW PRINT   'DEFORMATION
4460 ON ORTH GOTO 4590,4470
4470 OFSY=0
4480 OFSX=200
4490 LOCATE 12,29
4500 PRINT"DEFORMED PICTURE"
```

```
4510 GOSUB 6180'DEFORMED  KEY SCREENING.
4520 'TAKE INTO HBUF.DAT DEFORMED KEY.
4530 FOR I=1 TO XVECB
4540 LSET FHP$=MKS$(U(I))
4550 PUT #3,I
4560 NEXT I
4570 'G=40:H=88:GOSUB 31311    'FIRST REDUCTION.
4580 'OFSY=100:OFSX=200:GOSUB 30310 'AFTER PASS1 SCREENING.
4590 FRI=TVECS+1
4600 GOSUB 5660           'ORTH(X) Ç NSSP BULMA.
4610 'GET AGAIN DEFORMED KEY.
4620 FOR I=1 TO XVECB
4630 GET #3,I
4640 U(I)=CVS(FHP$)
4650 NEXT I
4660 'OFSY=100:OFSX=200:GOSUB 30310
4670 CLOSE #1,#2,#3
4680 ON ORTH GOTO 4690,6100      'FINDED KEY SCREENING.
4690 CLS
4700 LOCATE 3,1
4710 PRINT"üü H üü82=";
4720 'PRINT USING "##.##8888";NSSP
4730 PRINT"--------------------"
4740 FOR FRR=1 TO XVECB
4750 U(FRR)=U(FRR)-H(FRR)
4760 'IF U(FRR)>.5 THEN U(FRR)=1 ELSE U(FRR)=0
4770 PRINT"H(";FRR;")=";
4780 PRINT USING "##.##8888";U(FRR)
4790 NEXT FRR
4800 PRINT
4810 PRINT
4820 PRINT"PRESS ANY KEY TO CONTINUE."
4830 IF INKEY$<>"" THEN 4840 ELSE 4830
4840 RETURN
4850 PRINT"DEFORMED AREA %";DNY,
4860 ON DNC GOTO 4870,4920,4970
4870 PRINT"MODE : FILL"
4880 PRINT SFILLN;" PIXEL FILLED WITH ";DNWC
4890 GOSUB 5020
4900 PRINT"DEF %";SFILLN/3520*100
4910 RETURN
4920 PRINT"MODE : TOGGLE"
4930 PRINT"NUMBER OF TOGGLED PIXEL  :";STOGN
4940 GOSUB 5020
4950 PRINT"DEF %";STOGN/3520*100
4960 RETURN
4970 PRINT"MODE : OR"
4980 PRINT"NUMBER OF OR'ED PIXEL         :";SORN
4990 GOSUB 5020
5000 PRINT"DEF %";SORN/3520*100
```

```
5010 RETURN
5020 PRINT"NUMBER OF TOTAL PIXEL IS : 3520"
5030 RETURN
5040 '£££££££££££££££££ SUB SCALAR PRODUCT £££££££££££
5050 ISP=0
5060 FOR FRSC=1 TO XVECB
5070 ISP=ISP+U(FRSC)*V(FRSC)
5080 NEXT FRSC
5090 IF ISP>1 AND ISP<-1 THEN 5110
5100 ISP=INT(ISP*10000)/10000
5110 RETURN
5120 '£££££££ SUB SIMULATION CHECK FOR XVECS.DAT £££££££££££
5130 CLS
5140 IF FLAG=0 THEN PRINT"FILE=";REF1$:GOTO 5160
5150 PRINT"FILE=";REF2$
5160 PRINT
5170 ARG=TVECS*XVECB+2
5180 IF FLAG=1 THEN ARG=TVECS*XVECB+TVECS+2
5190 FOR FRCH=1 TO ARG
5200 IF FLAG=1 THEN GET #2,FRCH:PRINT CVS(FH$);"  ";:GOTO 5230
5210 GET #1,FRCH
5220 PRINT"(";FRCH;"=";CVS(FX$);") ";
5230 'PRINT:STOP
5240 NEXT FRCH
5250 PRINT
5260 FLAG=0
5270 RETURN
5280 '£££££££££ SUB ORTONORMAL VEC.METERS. £££££££££££££££
5290 CLS
5300 PRINT"FILE=";REF2$
5310 PRINT
5320 GET #2,1
5330 TVECS=CVS(FH$)
5340 ARI=XVECB+1
5350 ARL=1
5360 GET #2,2
5370 XVECB=CVS(FH$)
5380 FOR FRCH=3 TO TVECS*XVECB+2+TVECS
5390 GET #2,FRCH
5400 IF ARI>XVECB THEN 5460
5410 PRINT"H(";ARI;")=";
5420 PRINT USING "##.####";CVS(FH$)
5430 ARI=ARI+1
5440 NEXT FRCH
5450 RETURN
5460 PRINT
5470 PRINT USING "###";ARL;
5480 PRINT " . ORTHONORMAL VECTOR."
5490 PRINT"------------------------"
5500 PRINT"UU H UU82=";
```

```
5510 PRINT USING "##.##6666";CVS(FH$)
5520 PRINT
5530 ARI=1
5540 ARL=ARL+1
5550 GOTO 5440
5560 '£££££££££££££ SUB  X CEK. £££££££££££££££££££££
5570 'FRI koduna gore X cekilir.
5580 ARAAT1=(FRI-1)*XVECB+3
5590 I=1
5600 FOR FRX=ARAAT1 TO XVECB+ARAAT1-1
5610 GET #1,FRX
5620 U(I)=CVS(FX$)
5630 I=I+1
5640 NEXT FRX
5650 RETURN
5660 '£££££££££ SUB U(i) icinde verilen X icin £££££££££
5670 'üüorth(X)üü62 --> NSSP icine,
5680 'orth(X) --> H(I) icine konulur.
5690 FOR ICLR=1 TO XVECB
5700 H(ICLR)=0
5710 NEXT ICLR
5720 FOR FRJ=1 TO FRI-1
5730 ARAAT2=(FRJ-1)*XVECB+3+FRJ
5740 GET #2,ARAAT2-1
5750 NSH=CVS(FH$)
5760 IF NSH<=.0000001 AND NSH>=-.0000001 THEN 5880
5770 I=1
5780 FOR FRH=ARAAT2 TO XVECB+ARAAT2-1
5790 GET #2,FRH
5800 V(I)=CVS(FH$)
5810 I=I+1
5820 NEXT FRH
5830 GOSUB 5040        'SCALAR PRODUCT       X(I)*H(I).
5840 ITRSP=ISP
5850 FOR FRM=1 TO XVECB
5860 H(FRM)=H(FRM)+ITRSP*V(FRM)/NSH
5870 NEXT FRM
5880 NEXT FRJ
5890 FOR I=1 TO XVECB
5900 H(I)=U(I)-H(I)
5910 V(I)=H(I)
5920 U(I)=H(I)
5930 NEXT I
5940 GOSUB 5040         'SCALAR PRODUCT
5950 NSSP=ISP
5960 RETURN
5970 '£££££££££££ sub px cek. £££££££££££££££££££££££
5980 MM=(FRI-1)*577+5
5990 I=1
6000 FOR FRXX=MM TO MM+439
```

```
6010 GET #1,FRXX
6020 S=ASC(FP$)
6030 FOR J=7 TO 0 STEP -1
6040 U(I)=2@J AND S
6050 IF U(I)=>1 THEN U(I)=1 ELSE U(I)=0
6060 I=I+1
6070 NEXT J
6080 NEXT FRXX
6090 RETURN
6100 'SUB SCREENING.....................................
6110 LOCATE 12,55
6120 PRINT"FINAL PICTURE"
6130 OFSY=0
6140 OFSX=400
6150 FOR I=1 TO XVECB
6160 U(I)=U(I)-H(I)
6170 NEXT I
6180 X=1
6190 Y=1
6200 LINE(OFSX,OFSY)-(OFSX+177,OFSY+81),,BF
6210 FOR I=1 TO XVECB
6220 IF U(I)>=.5 THEN U(I)=1 ELSE U(I)=0
6230 IF U(I)=0 THEN 6280
6240 PRESET(OFSX+Y*2,OFSY+X*2)
6250 PRESET(OFSX+Y*2-1,OFSY+X*2-1)
6260 PRESET(OFSX+Y*2,OFSY+X*2-1)
6270 PRESET(OFSX+Y*2-1,OFSY+X*2)
6280 Y=Y+1
6290 IF Y>88 THEN X=X+1:Y=1
6300 NEXT I
6310 'OFSX=0:OFSY=0
6320 LOCATE 14,25
6330 PRINT"PRESS ANY KEY TO CONTINUE."
6340 IF INKEY$<>"" THEN 6350 ELSE 6340
6350 LOCATE 14,25
6360 PRINT"                              "
6370 RETURN
6380 '************************ SUB FIRST ERROR REDUCTION.............
6390 'N=3:M=3
6400 'FOR GI=1 TO N
6410 'FOR GJ=1 TO M
6420 'READ OP(GI,GJ)
6430 'PRINT OP(GI,GJ);
6440 'NEXT GJ,GI
6450 'DATA 0,-.5,0,-.5,3,-.5,0,-.5,0
6460 FOR FRL=0 TO G-N
6470 FOR FRK=0 TO H-M
6480 TOP=0
6490 FOR I=1 TO N
6500 FOR J=1 TO M
```

```
6510 TOP=TOP+OP(I,J)*U(FRK+J+(FRL+I-1)*H)
6520 NEXT J,I
6530 U(FRK+1+FRL*H)=TOP
6540 NEXT FRK,FRL
6550 'ENBYK=-1000
6560 'FOR I=1 TO XVECB
6570 'IF U(I)>ENBYK THEN ENBYK=U(I)
6580 'NEXT I
6590 'IF ENBYK=0 THEN 31337
6600 FOR I=1 TO XVECB
6610 'U(I)=U(I)/ENBYK
6620 IF U(I)>=.5 THEN U(I)=1 ELSE U(I)=0
6630 NEXT I
6640 RETURN
6650 '#########################SUB DEFORMATION
6660 VIEW PRINT 19 TO 24
6670 PRINT"IS THERE ANY DEFORMATION ?  (Y/N) :";
6680 YN$=INKEY$
6690 IF YN$="" THEN 6680 ELSE PRINT YN$
6700 IF YN$="Y" OR YN$="y" THEN 6720
6710 IF YN$="N" OR YN$="n" THEN 7040 ELSE 6680
6720 RANDOMIZE TIMER
6730 GOSUB 7070                                  'CLEAR.
6740 LOCATE 19,1
6750 INPUT"AMOUNT OF THE DEFORMATION IN PERCENT(Max.=100) :",DNY
6760 GOSUB 7070                                  'CLEAR.
6770 IF DNY>100 THEN BEEP:GOTO 6740
6780 IF DNY=<0 THEN BEEP:GOTO 7050
6790 DNM=INT(DNY*XVECB/100)
6800 PRINT"BEGINNING OF THE DEFORMATION"
6810 PRINT"1. RANDOM"
6820 PRINT"2. MANUEL"
6830 INPUT"SELECT ONE :",DNB
6840 GOSUB 7070                                  'CLEAR
6850 IF DNB<1 OR DNB>2 THEN BEEP:GOTO 6800
6860 IF INT(DNB)=2 THEN 6890
6870 DNBD=INT((XVECB-DNM)*RND)
6880 GOTO 6920
6890 'LOCATE 19,1
6900 PRINT"START NUMBER OF DEFORMATION .(Max.:";XVECB-DNM;") :";
6910 INPUT"",DNBD
6920 GOSUB 7070                                  'CLEAR.
6930 IF (XVECB-DNM)<DNBD THEN DNBD=XVECB-DNM
6940 PRINT"WHICH DO YOU WANT TO USE THIS TECHNIQUES IN THE DEFORMATION"
6950 PRINT"1. ERASE  FRAGMENT"
6960 PRINT"2. TOGGLE  FRAGMENT"
6970 PRINT"3. LOGIC .OR. FRAGMENT"
6980 'LOCATE 19,1
6990 INPUT"SELECT ONE :",DNC
7000 GOSUB 7070                                  'CLEAR
```

```
7010 IF DNC<1 OR DNC>3 THEN BEEP:GOTO 6940
7020 ON INT(DNC) GOSUB 7130,7270,7490
7030 RETURN
7040 GOSUB 7070                                  'CLEAR.
7050 RETURN
7060 'SUB CLEAR ================================
7070 PRINT
7080 PRINT
7090 PRINT
7100 PRINT
7110 PRINT
7120 RETURN
7130 'SUB SILME ================================
7140 PRINT"ERASING FRAGMENT WILL BE"
7150 PRINT"0. FILLED WITH 0"
7160 PRINT"1. FILLED WITH 1"
7170 INPUT"SELECT ONE :",DNWC
7180 GOSUB 7070                                  'CLEAR.
7190 DNWC=INT(DNWC)
7200 IF DNWC<0 OR DNWC>1 THEN BEEP:GOTO 7130
7210 SFILLN=0
7220 FOR FRD=DNBD TO DNBD+DNM
7230 U(FRD)=DNWC:SFILLN=SFILLN+1
7240 NEXT FRD
7250 SFILLC=DNWC
7260 RETURN
7270 'SUB DEF. TOGGLE ==========================
7280 PRINT"TOGGLED FRAGMENT WITH"
7290 PRINT"1. COMPLETELY"
7300 PRINT"2. RANDOM,"
7310 'LOCATE 19,1
7320 INPUT "SELECT ONE :",DNWT
7330 GOSUB 7070                                  'CLEAR.
7340 DNWT=INT(DNWT)
7350 IF DNWT<1 OR DNWT>2 THEN BEEP:GOTO 7270
7360 ON DNWT GOTO 7370,7430
7370 STOGN=0
7380 FOR FRD=DNBD TO DNBD+DNM
7390 U(FRD)=U(FRD) XOR 1
7400 STOGN=STOGN+1
7410 NEXT FRD
7420 RETURN
7430 STOGN=0
7440 FOR FRD=DNBD TO DNBD+DNM
7450 IF RND>=.5 THEN U(FRD)=U(FRD) XOR 1:STOGN=STOGN+1:GOTO 7470
7460 U(FRD)=U(FRD) XOR 0
7470 NEXT FRD
7480 RETURN
7490 'SUB LOJIK OR ILE DEF. ====================
7500 SORN=0
```

```
7510 FOR FRD=DNBD TO DNBD+DNM
7520 IF RND>=.5 THEN SORN=SORN+1:U(FRD)=U(FRD) OR 1
7530 NEXT FRD
7540 RETURN
7550 'SUB TVECS SETÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇÇ
7560 OPEN "R",#2,REF2$,4
7570 FIELD #2,4 AS FH$
7580 LSET FH$=MKS$(ITVECS)
7590 PUT #2,1
7600 CLOSE #2
7610 RETURN
```

EK-B PROGRAM KULLANIM ÖRNEKLERİ

Örnek 1 : Vektör uzayının her bir elemanın, kullanıcı tarafından tek tek girilmesi yoluyla tanımlanarak kütük içine yerleştirilmesi.

Işlem : 1 numaralı kod seçilir. 1 nolu kod ile MANUEL SIMULATION'a geçmeden önce 5 nolu kod ile simulasyon sonucunun konulacağı kütük bilgileri tanımlanmalıdır. (bkz.örnek 3)

Birinci ekran :

```
.............MAIN MENU.............
1.   MANUEL SIMULATION
2.   AUTOMATIC SIMULATION
3.   ORTHONORMALIZATION
4.   RECALL
5.   FILE UTILITIES
6.   COLOR
7.   HELP
8.   FIRST REDUCTION
9.   END OF PROGRAM
SELECT ONE : 1
```

İkinci ekran :

-- Simulasyon kütük bilgileri kontrol edilir.

```
NUMBER OF TOTAL PATTERN: 25
LENGHT OF EACH  PATTERN: 3520
IS IT CORRECT ?  (Y/N) :
```

N : seçimi ile ; MAIN MENU'ye dönülür ve istenirse öncelikle kütük bilgileri örnek 3 'deki gibi değiştirilebilir.

Y : seçimi ile ; referans kütük içine belirtilen örüntüler interaktif olarak oluşturulur.

Örnek 2 : Vektör uzayının bilgisayar tarafından rassal olarak oluşturularak bir kütük üzerine yerleştirilmesi. Vektör sayısı ve herbir vektör elemanının alacağı değer aralığı önceden verilecektir.

İşlem : 2 numaralı kod seçilir. 2 nolu kod ile AUTOMATIC SIMULATION'a geçmeden önce 5 nolu kod ile simulasyon sonucunun konulacağı kütük bilgileri tanımlanmalıdır.(Bakınız örnek 3)

Birinci ekran :

```
.............MAIN MENU.............
1.  MANUEL SIMULATION
2.  AUTOMATIC SIMULATION
3.  ORTHONORMALIZATION
4.  RECALL
5.  FILE UTILITIES
6.  COLOR
7.  HELP
8.  FIRST REDUCTION
9.  END OF PROGRAM
SELECT ONE : 2
```

İkinci ekran :

-- Simulasyon kütük bilgileri kontrol edilir.

```
NUMBER OF TOTAL PATTERN: 25
LENGHT OF EACH  PATTERN: 3520
IS IT CORRECT ?  (Y/N) :
```

N : seçimi ile ; MAIN MENU'ye dönülür ve istenirse öncelikle kütük bilgileri örnek 3 'deki gibi değiştirilebilir.
Y : seçimi ile ; referans kütük içine belirtilen sayıda ve uzunluktaki örüntü bilgisayar tarafından rassal olarak doldurulur. Herbir elemanın hangi aralıktaki değerde olacağı işlem esnasında istenilmektedir.

Örnek 3 : Simulation için kütük bilgilerinin tanımlanması. Örneğin içine herbirinin boyu 3520 olan 25 örüntü lü OVECS.REF isimli kütüğü simulasyon işleminde kullanmak üzere set etmek için şunlar yapılır.

MAIN MENU 'den 5 nolu kod seçilir ve ikinci ekrandan (aşağıda) 3 nolu kod seçilir ve aşağıdaki işlemler yapılır.

İkinci ekran :

.....FILE UTILIES.....

1. LIST REFERANCE FILES
2. LIST ORTHONORMAL FILES
3. SET FILE PARAMETERS
4. CREATE DRAWING
5. RETURN TO MAIN MENU

SELECT ONE : 3

REFERANCE FILE NAME default <SLIB.REF> ENTER
ORTHONORMAL FILE NAME default <HPVECS1.REF> ENTER
SIMULATION REF. FILE NAME default <XVECS.REF> OVECS.REF

NUMBER OF TOTAL SIMULATION PATTERN default < 3 > 25
LENGHT OF EACH SIMULATION PATTERN default < 3520 > ENTER

ACTIVE ORTHONORMAL VECTOR NUMBER IN FILE default < 3 > ENTER

Örnek 6 : Verilen referans vektör uzayındaki istenilen sayıdaki vektörün ortonormalleştirilerek yeni bir kütüğe yerleştirilmesi.

Ortonormalleştirme için; MAIN MENU 'DEN 3 nolu KOD seçilir.3 nolu kod ile ORTHONORMALIZATION 'a geçmeden önce 5 nolu kod ile ortonormalleştirme için gerekli kütük bilgileri (eğer default uygun değil ise) tanımlanmalıdır.
(bkz.örnek 4)

İkinci ekran :

--Ortonormalizasyon kütük bilgileri kontrol edilir.

REFERANCE PATTERN FILE NAME:SLIB.REF

ORTHONORMAL FILE NAME:HPVEC1.ORT

IS IT CORRECT ? (Y/N) :

N : Seçimi ile MAIN MENU 'ye dönülür. İstenirse kütük bilgileri örnek 4 'teki gibi değiştirilebilir.

Y : Seçimi ile referans kütükten alınan vektörler ortonormalleştirme işleminden geçirilerek ortonormal kütüğe yerleş-

tirilirler. (Ortonormalleştirilecek vektör sayısı örnek 4 te gösterildiği gibi tanımlanır.)

Örnek 4 : ORTHONORMALIZATION esnasında kullanılacak kütük bilgilerinin tanımlanması. Örneğin ; ORTHONORMALIZATION ön cesinde :

-- Ortonormal hale getirilecek vektörleri saklayan referans kütük ismi : ULIB.REF ,

-- Ortonormal hale getirilen vektörlerin saklanacağı kütük ismi : HPVEC.ORT ,

-- Ortonormalleştirilecek vektör sayısıda 5 olsun.(ULIB.REF kütuğünün ilk 5 vektörü işleme alınacaktır.)

Öncelikle MAIN MENU 'den 5 nolu kod ile aşağıda görülen FILE UTILIES alt menüsüne geçilir. Buradan 3 nolu kodla beraber aşağıdaki tanımlamalar yapılır.

İkinci ekran :

```
.....FILE UTILIES.....
1.  LIST REFERANCE FILES
2.  LIST ORTHONORMAL FILES
3.  SET FILE PARAMETERS
4.  CREATE DRAWING
5.  RETURN TO MAIN MENU
SELECT ONE : 3
```

```
REFERANCE FILE NAME default <SLIB.REF>  ULIB.REF
ORTHONORMAL FILE NAME default <HPVECS1.REF> HPVEC.ORT
SIMULATION REF. FILE NAME default <XVECS.REF> ENTER

NUMBER OF TOTAL SIMULATION PATTERN default < 3 > ENTER
LENGHT OF EACH  SIMULATION PATTERN default < 3520 > ENTER

ACTIVE ORTHONORMAL VECTOR NUMBER IN FILE default < 3 > 5
```

Örnek 7 : Kullanıcının tanımladığı bir anahtar vektörün yine kullanıcı tarafından belirlenen ortonormal uzay üzerine izdüşümünün alınması. Gösterilen ortonormal uzayda istenilen sayıdaki vektör üzerine izdüşüm alınabilir.

RECALL için; MAIN MENU 'DEN 4 nolu KOD seçilir.4 nolu kod ile RECALL 'a geçmeden önce 5 nolu kod ile RECALL

için gerekli kütük bilgileri (eğer default uygun değil ise)

tanımlanmalıdır. (bkz.Örnek 5)

Ikinci ekran :

--RECALL işleminin hangi referans kütükte yapılacağı belirlenir.

RECALL

RECALL : 1. in USER FILE
2. in PMASTER FILE
SELECT ONE :

1 kodunun seçilmesi halinde sonuç vektörünün herbir elemanının değeri görülür.
2 kodunun seçilmesi halinde vektör değerleri ekrana bastırılır. Eğer anahtar vektör veya ortonormal uzay resim olarak bir anlam ifade etmiyorsa 2 kodu seçilmez.

Örnek 4 : RECALL esnasında kullanılacak kütük bilgilerinin tanımlanması. Örneğin ; RECALL öncesinde :

-- RECALL için anahtar vektör simülasyonu esnasında kullanılacak kütük ismi : ULIB.REF ,
-- RECALL için üzerine ortonormal izdüşüm alınacak ortonormal kütük ismi : HPVEC.ORT ,
-- RECALL için kullanılacak aktif ortonormal kütük üzerindeki vektör sayısıda 5 olsun.

Öncelikle MAIN MENU 'den 5 nolu kod ile aşağıda görülen FILE UTILIES alt menüsüne geçilir. Buradan 3 nolu kodla beraber aşağıdaki tanımlamalar yapılır.

Ikinci ekran :

.....FILE UTILIES.....
1. LIST REFERANCE FILES
2. LIST ORTHONORMAL FILES
3. SET FILE PARAMETERS
4. CREATE DRAWING
5. RETURN TO MAIN MENU
SELECT ONE : 3

REFERANCE FILE NAME default <SLIB.REF> ULIB.REF
ORTHONORMAL FILE NAME default <HPVECS1.REF> HPVEC.ORT
SIMULATION REF. FILE NAME default <XVECS.REF> ENTER

NUMBER OF TOTAL SIMULATION PATTERN default < 3 > ENTER
LENGHT OF EACH SIMULATION PATTERN default < 3520 > ENTER

ACTIVE ORTHONORMAL VECTOR NUMBER IN FILE default < 3 > 5

Örnek 8 : Ekran renk kodlarının değiştirilmesi.

MAIN MENU 'den 6 nolu kod seçilir. Daha sonra aşağıda ki listede verilen renkler seçilebilir. Örneğin ;

-- kalem rengi **YEŞIL**,

-- zemin rengi **KIRMIZI** yapılmak istenirse,

İkinci ekran :

```
..........................COLOR.................................
0   BLACK                          8   GRAY
1   BLUE                           9   LIGHT BLUE
2   GREEN                         10   LIGHT GREEN
3   CYAN                          11   LIGHT CYAN
4   RED                           12   LIGHT RED
5   MAGENTA                       13   LIGHT MAGENTA
6   BROWN                         15   YELLOW
7   WHITE                         16   HIGH INTENSITY WHITE
```

FOREGROUND COLOR default < 7 > **2**

BACKGROUND COLOR default < 4 > **1**

yapılmalıdır.

Örnek 9 : Komutlar hakkında açıklayıcı bilgiler alınması. MAIN MENU 'den 7 nolu kodun seçilir. Sonuçta herbir kodun kullanımı hakkında açıklayıcı bilgiler ekrana gelir.

Örnek 10 : Programından çıkma. Bu işlem için MAIN MENU 'den 9 nolu kod seçilir. İşlem sonucunda **DOS** işletim sistemine geçilir.

Örnek 11 : FIRST REDUCTION (F.R.) işleminin uygulanması. MAIN MENU 'den 8 nolu kod ile aşağıdaki ekrana geçilir.

```
LAPLACIAN              CONVOLUTION            USER DEF

  0 -.5   0            1   1   1              -   -   -
-.5   3 -.5            1   1   1              -   -   -
  0 -.5   0            1   1   1              -   -   -
```

1. LAPLACIAN
2. MEAN
3. CONVOLUTION
4. USER DEFINED CORE
5. FIRST REDUCTION FLAG OFF

SELECT ONE :

Burada ;
1. koduyla LAPLACIAN işlemi,
2. koduyla MEAN işlemi,
3. koduyla CONVOLUTION işlemi,
4. koduylada kullanıcının tanımlayacağı çekirdek seçilmiş olur.

Yukarıdaki kodlardan herhangibirinin seçilmesi halinde F.R. flaği set edilir. Yani ORTHONORMALIZATION ve RECALL esnasında F.R. işleminde yapılacağı işaret edilmiş olur.

5. kod ile F.R. flaği RESET edilir. ORTHONORMALIZATION ve RECALL esnasında F.R. işleminin yapılmayacağını bildirir.

Örnek 12 : Görüntü şeklindeki referans kütüğün kullanıcı tarafından oluşturulması.
MAIN MENU 'den 5 nolu kod ile FILE UTILIES alt menüsüne geçtikten sonra 4 nolu kod seçilir. Böylece grafik editöre geçilmiş olacaktır. Kullanıcı GRAFIK EDITOR yardımıyla şekli oluşturur.

EK-C MATEMATIKSEL ORTOGONALLEŞTIRME SAYISAL ÖRNEĞI

Referans vektörler.

$x_1=(1,0,2,5)$ - Toplam 3 referans vektör
$x_2=(3,4,7,8)$ - her bir vektörün boyu 4
$x_3=(9,7,6,5)$ olsun.

Verilmiş olan bu referans vektörleri Gram-Schmidth işlevi yardımıyla ortogonal hale getirelim.

$$h_1=x_1=(1,0,2,5)$$

$$h_2=x_2-\langle x_2,h_1\rangle\frac{h_1}{norm(h_1)^2}$$

$$h_3=x_3-\langle x_3,h_1\rangle\frac{h_1}{norm(h_1)^2}-\langle x_3,h_2\rangle\frac{h_2}{norm(h_2)^2}$$

$$\langle x_2,h_1\rangle=(3,4,7,8)(1,0,2,5)=3+14+40=57$$

$$norm(h_1)^2=1+4+25=30$$

$$h_2=(3,4,7,8)-\frac{57}{30}(1,0,2,5)$$

$$h_2=(1.1,4,3.2,-1.5)$$

$$norm(h_2)^2=1.21+16+10.24+2.25=29.7$$

$$\langle x_3,h_1\rangle=(9,7,6,5)(1,0,2,5)=9+0+12+25=46$$

$$\langle x_3,h_2\rangle=(9,7,6,5)(1.1,4,3.2,-1.5)$$
$$=9.9+28+19.2-7.5=49.6$$

$$h_3=(9,7,6,5)-\frac{46}{30}(1,0,2,5)-\frac{49.6}{29.7}(1.1,4,3.2,-1.5)$$

$$h_3=((9-1.533-1.837),(7,0,6.68),(6-3.067-5.344),(5-7.667+2.505))$$

$$h_3=(5.63,0.32,-2.411,-0.162)$$

$$norm(h_3)^2=37.638$$

Eğer bu işlemler program yardımı ile yapılsaydı, referans vektörler ve ortonormal vektörlerin kütük içindeki dizimleri şöyle olacaktı,

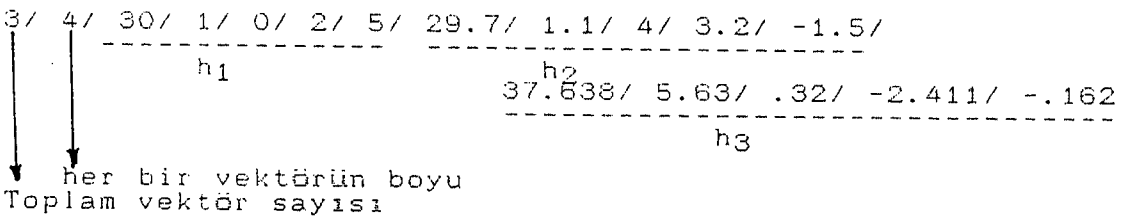
- Referans vektörler için:

3 / 4 / 1 / 0 / 2 / 5 / 3 / 4 / 7 / 8 / 9 / 7 / 6 / 5

(1 / 0 / 2 / 5 = x_1; 3 / 4 / 7 / 8 = x_2; 9 / 7 / 6 / 5 = x_3)

↓ Toplam vektor sayısı
↓ her bir vektörün boyu

- Ortonormal vektörler için :



EK-D.1.

Referans kütük ismi : SLIB.REF

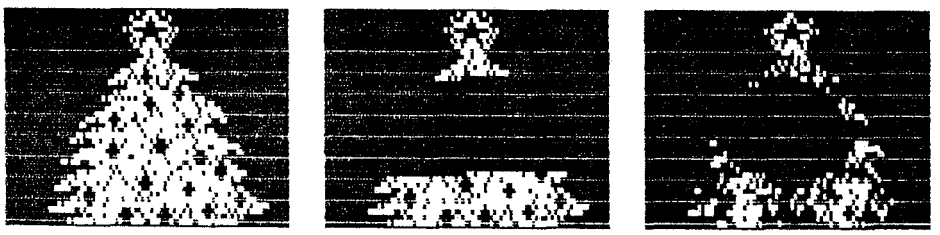
Referans kütükteki mevcut örüntü sayısı : 120

Referans kütükte kullanılan aktif örüntü sayısı : 120

Ortonormal kütük ismi : HPVECS1.ORT

Ortonormal kütükteki mevcut örüntü sayısı : 5

Ortonormal kütükte kullanılan aktif örüntü sayısı : 5



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 45          MODE : FILL
 1585  PIXEL FILLED WITH  0
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 45
```

ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

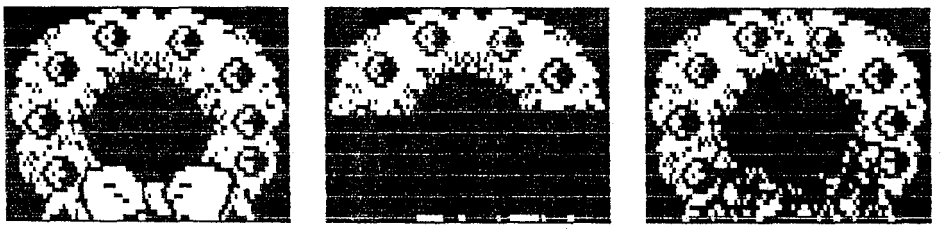
PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 100          MODE : OR
NUMBER OF OR'ED PIXEL          : 1785
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 50
```

ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

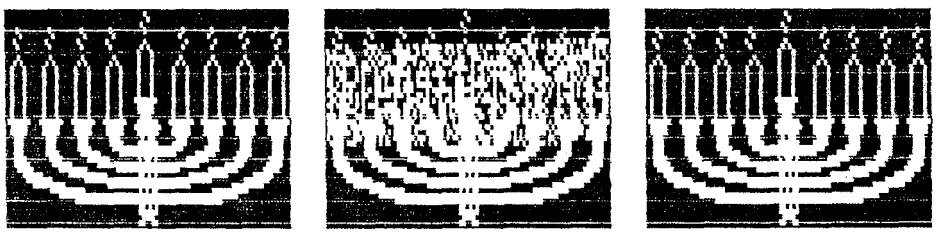
```
DEFORMED AREA % 100          MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 1786
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 50
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

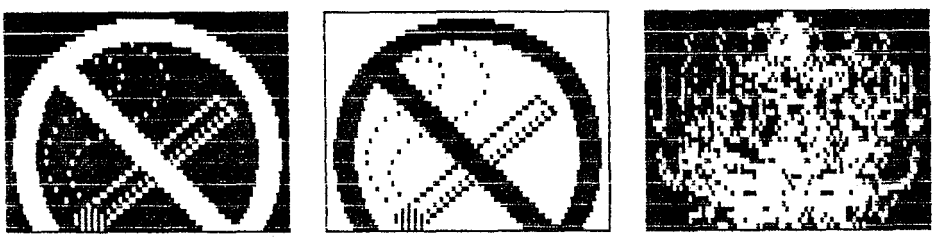
```
DEFORMED AREA % 50          MODE : FILL
 1761  PIXEL FILLED WITH  0
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 50.
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

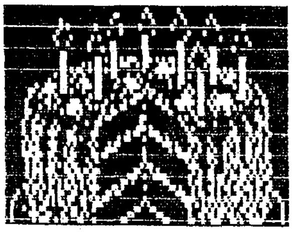
```
DEFORMED AREA % 48          MODE : OR
NUMBER OF OR'ED PIXEL         : 825
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 23
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

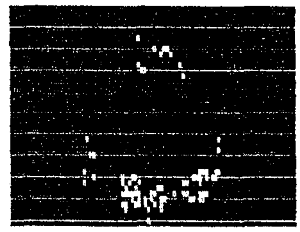
PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 100         MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 3521
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 100
```



ORIGINAL PICTURE

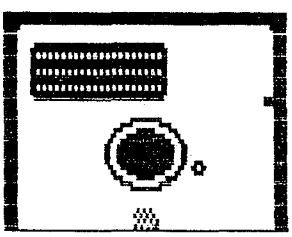
DEFORMED PICTURE



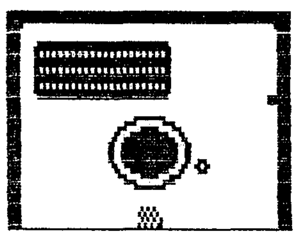
FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

DEFORMED AREA MODE :
NUMBER OF TOGGLED PIXEL :
NUMBER OF TOTAL PIXEL IS : 3520
DEF



ORIGINAL PICTURE

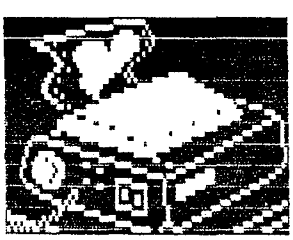


DEFORMED PICTURE

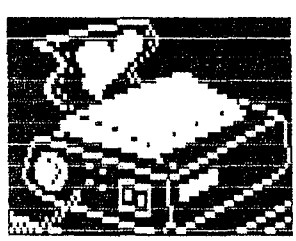
FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

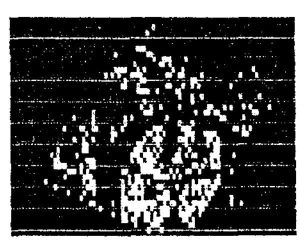
DEFORMED AREA MODE :
NUMBER OF TOGGLED PIXEL :
NUMBER OF TOTAL PIXEL IS : 3520
DEF



ORIGINAL PICTURE



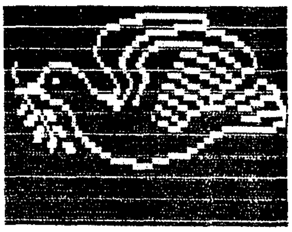
DEFORMED PICTURE



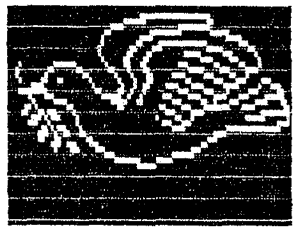
FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

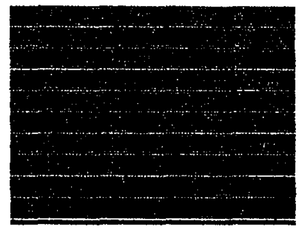
DEFORMED AREA MODE :
NUMBER OF TOGGLED PIXEL :
NUMBER OF TOTAL PIXEL IS : 3520
DEF



ORIGINAL PICTURE



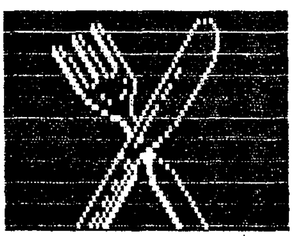
DEFORMED PICTURE



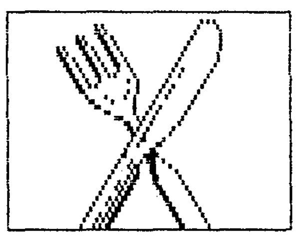
FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

DEFORMED AREA MODE :
NUMBER OF TOGGLED PIXEL :
NUMBER OF TOTAL PIXEL IS : 3520
DEF



ORIGINAL PICTURE



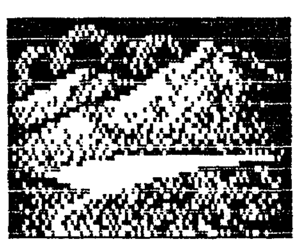
DEFORMED PICTURE

FINAL PICTURE

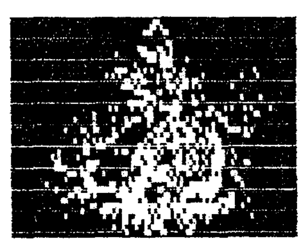
PRESS ANY KEY TO CONTINUE.

DEFORMED AREA MODE :
NUMBER OF TOGGLED PIXEL :
NUMBER OF TOTAL PIXEL IS : 3520
DEF

ORIGINAL PICTURE



DEFORMED PICTURE



FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

DEFORMED AREA MODE :
NUMBER OF TOGGLED PIXEL :
NUMBER OF TOTAL PIXEL IS : 3520
DEF

EK-D.2.

Referans kütük ismi : PATTERN1.REF

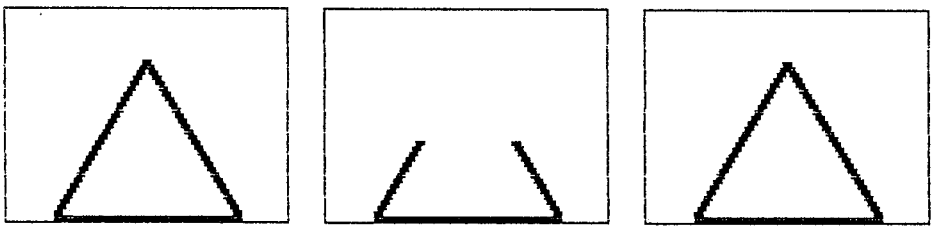
Referans kütükteki mevcut örüntü sayısı : 10

Referans kütükte kullanılan aktif örüntü sayısı : 10

Ortonormal kütük ismi : PATTERN1.ORT

Ortonormal kütükteki mevcut örüntü sayısı : 8

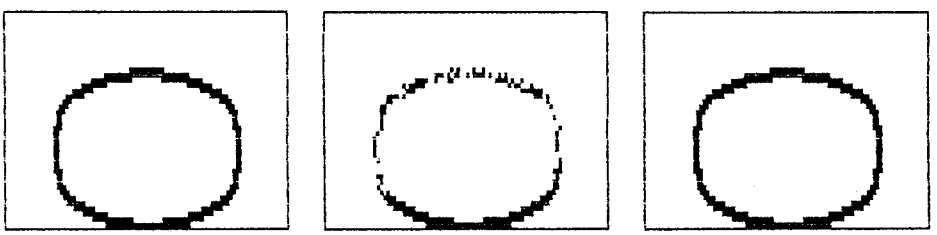
Ortonormal kütükte kullanılan aktif örüntü sayısı : 8



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

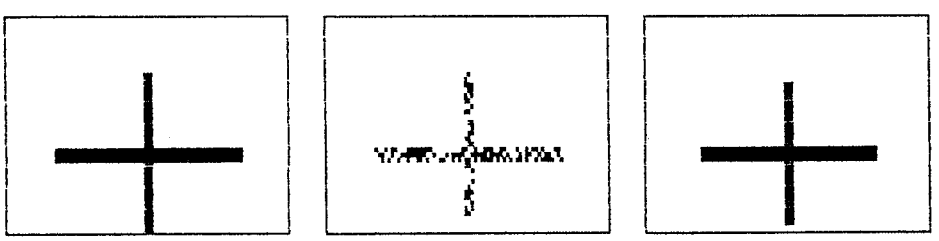
```
DEFORMED AREA % 60          MODE : FILL
 2113  PIXEL FILLED WITH  1
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 60.02841
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 75          MODE : OR
NUMBER OF OR'ED PIXEL         : 1335
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 37.92614
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 100         MODE : OR
NUMBER OF OR'ED PIXEL         : 1767
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 50.19886
```

ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

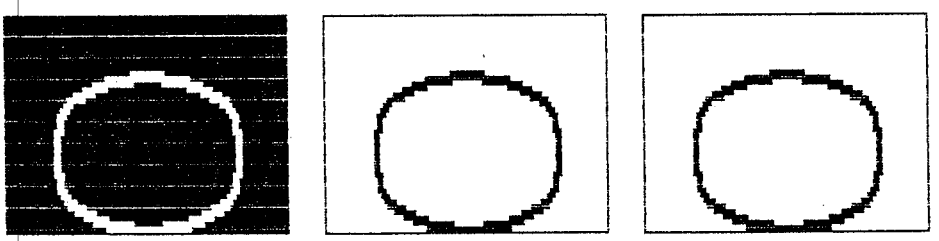
PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 100          MODE : OR
NUMBER OF OR'ED PIXEL          : 1819
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 51.67614
```

ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

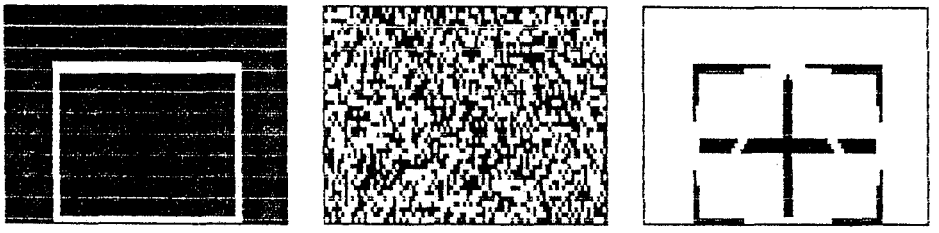
```
DEFORMED AREA % 50           MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 899
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 25.53977
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 100          MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 3521
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 100
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

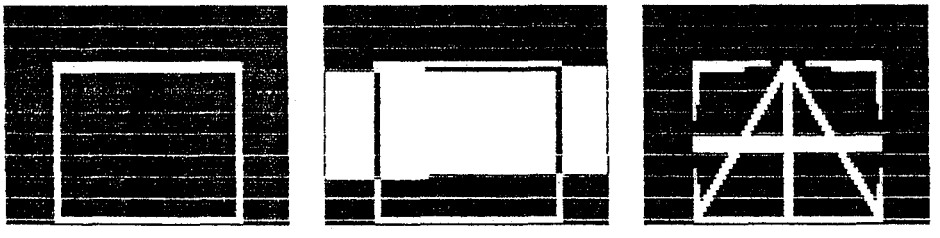
PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 100          MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 1792
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 50.90909
```

ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

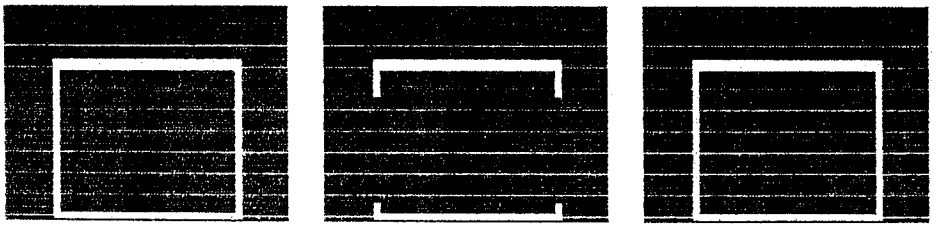
```
DEFORMED AREA % 100          MODE : OR
NUMBER OF OR'ED PIXEL        : 1767
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 50.19886
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

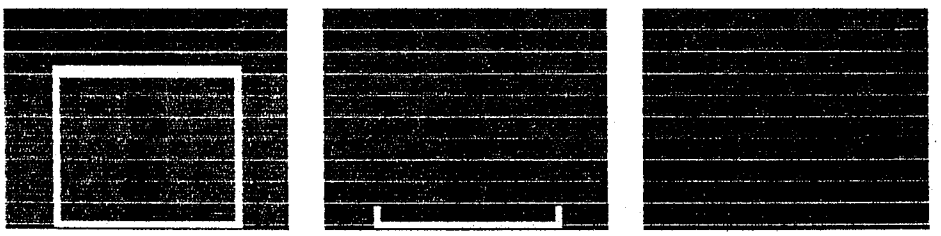
```
DEFORMED AREA % 50           MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 1761
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 50.02841
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

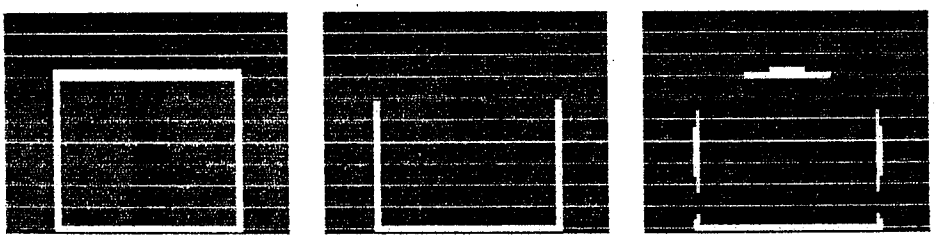
```
DEFORMED AREA % 50          MODE : FILL
 1761  PIXEL FILLED WITH  0
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 50.02841
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

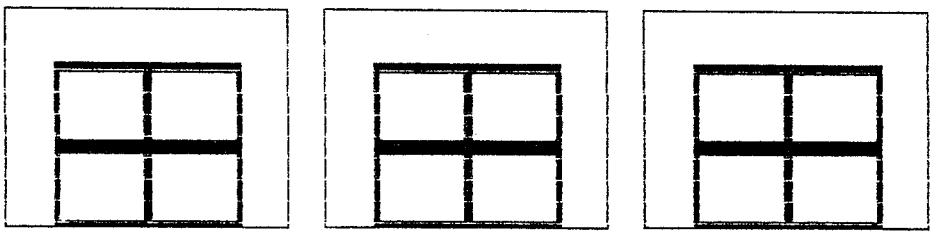
```
DEFORMED AREA % 90          MODE : FILL
 3169  PIXEL FILLED WITH  0
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 90.02841
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

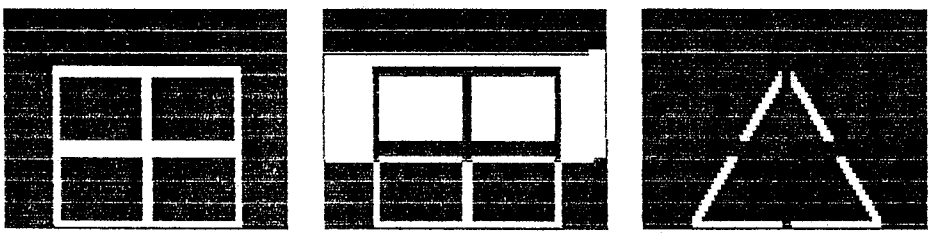
```
DEFORMED AREA % 40          MODE : FILL
 1409  PIXEL FILLED WITH  0
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 40.02841
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 0           MODE : FILL
 0  PIXEL FILLED WITH  0
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 0
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 50          MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 1761
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 50.02841
```

ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 100         MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 1724
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 48.97727
```

EK-D.3.

Referans kütük ismi : PATTERN2.REF

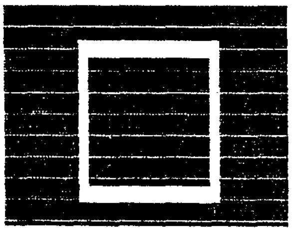
Referans kütükteki mevcut örüntü sayısı : 8

Referans kütükte kullanılan aktif örüntü sayısı : 8

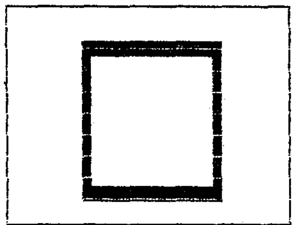
Ortonormal kütük ismi : PATTERN2.ORT

Ortonormal kütükteki mevcut örüntü sayısı : 8

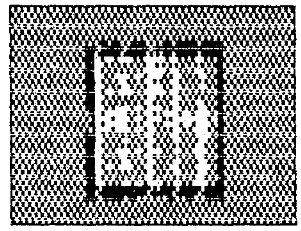
Ortonormal kütükte kullanılan aktif örüntü sayısı : 8



ORIGINAL PICTURE



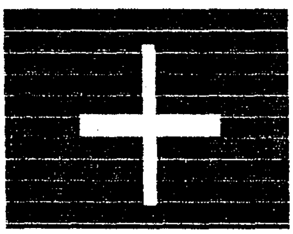
DEFORMED PICTURE



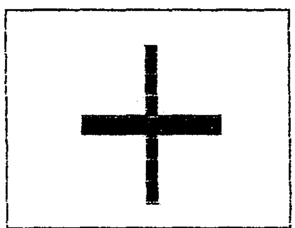
FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

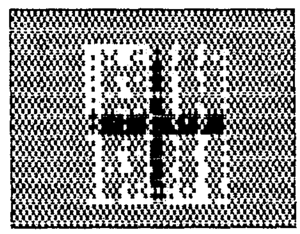
DEFORMED AREA % 100   MODE : TOGGLE
NUMBER OF TOGGLED PIXEL : 3521
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 100



ORIGINAL PICTURE



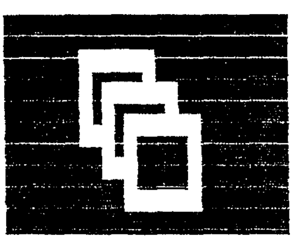
DEFORMED PICTURE



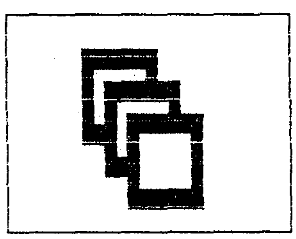
FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

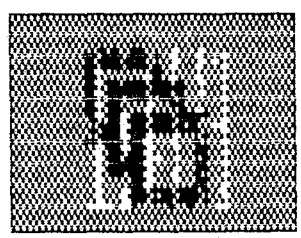
DEFORMED AREA % 100   MODE : TOGGLE
NUMBER OF TOGGLED PIXEL : 3521
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 100



ORIGINAL PICTURE



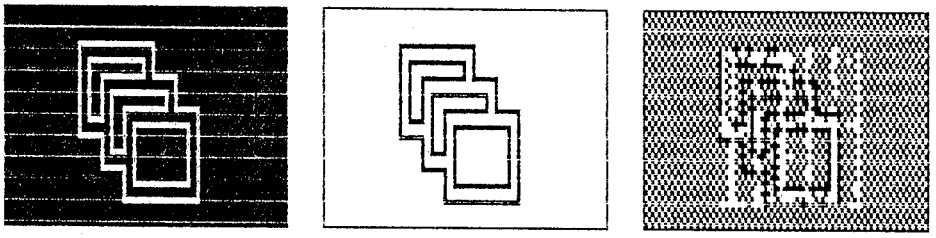
DEFORMED PICTURE



FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

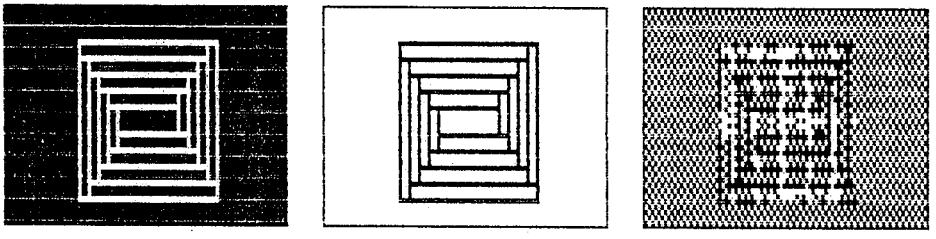
DEFORMED AREA % 100   MODE : TOGGLE
NUMBER OF TOGGLED PIXEL : 3521
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 100



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

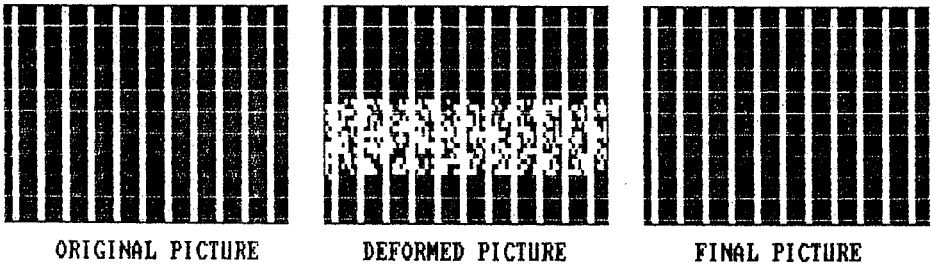
```
DEFORMED AREA % 100          MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 3521
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 100
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

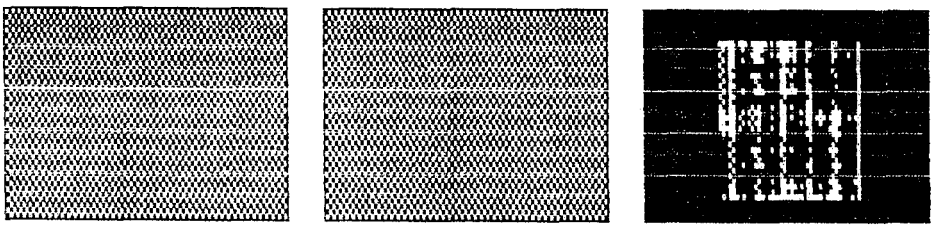
```
DEFORMED AREA % 100          MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 3521
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 100
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 35           MODE : OR
NUMBER OF OR'ED PIXEL        : 622
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 17.67045
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

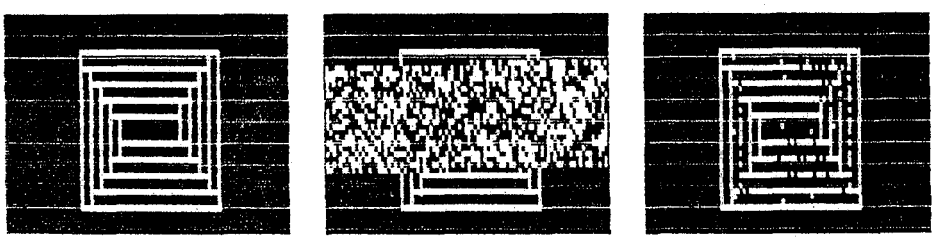
PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 100          MODE : TOGGLE
NUMBER OF TOGGLED PIXEL   : 3521
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 100
```

ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 50           MODE : TOGGLE
NUMBER OF TOGGLED PIXEL   : 874
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 24.82955
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 50           MODE : TOGGLE
NUMBER OF TOGGLED PIXEL   : 872
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 24.77273
```

EK-D.4.

Referans kütük ismi : PATTERN2.REF

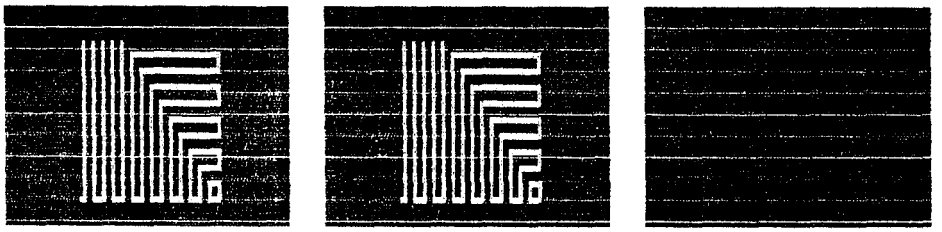
Referans kütükteki mevcut örüntü sayısı : 8

Referans kütükte kullanılan aktif örüntü sayısı : 8

Ortonormal kütük ismi : PATTERN1.ORT

Ortonormal kütükteki mevcut örüntü sayısı : 8

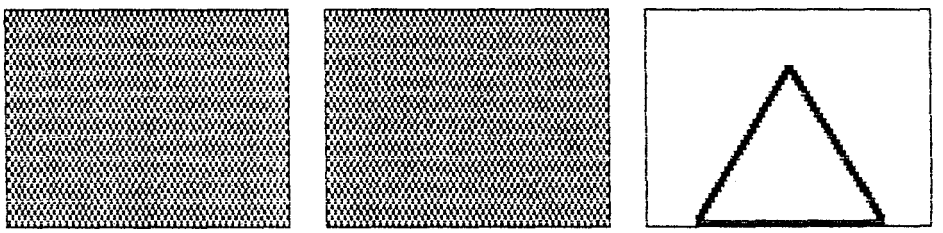
Ortonormal kütükte kullanılan aktif örüntü sayısı : 8



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

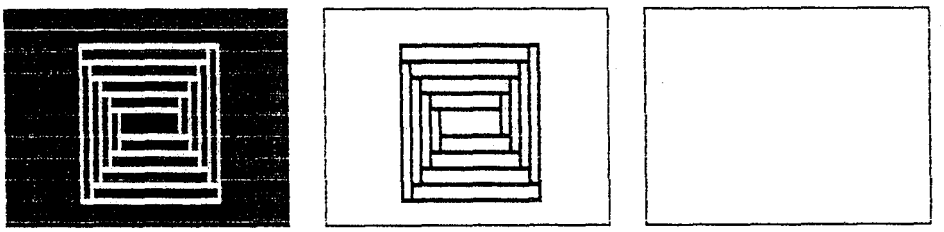
```
DEFORMED AREA                MODE :
NUMBER OF OR'ED PIXEL          :
NUMBER OF TOTAL PIXEL IS : 3520
DEF
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA                MODE :
NUMBER OF OR'ED PIXEL          :
NUMBER OF TOTAL PIXEL IS : 3520
DEF
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 100          MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 3521
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 100
```

EK-D.5.

Referans kütük ismi : PATTERN1.REF

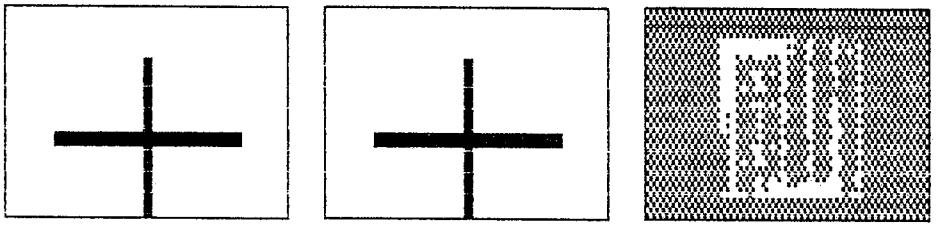
Referans kütükteki mevcut örüntü sayısı : 10

Referans kütükte kullanılan aktif örüntü sayısı : 10

Ortonormal kütük ismi : PATTERN2.ORT

Ortonormal kütükteki mevcut örüntü sayısı : 8

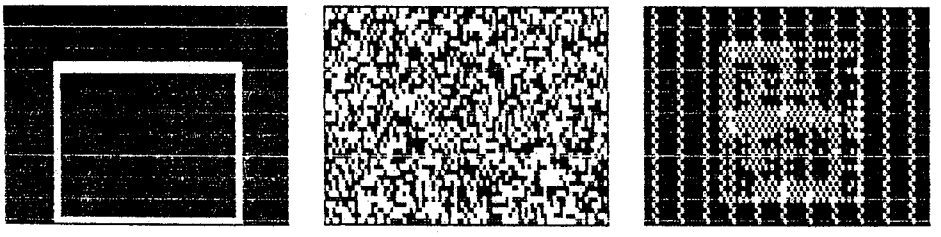
Ortonormal kütükte kullanılan aktif örüntü sayısı : 8



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

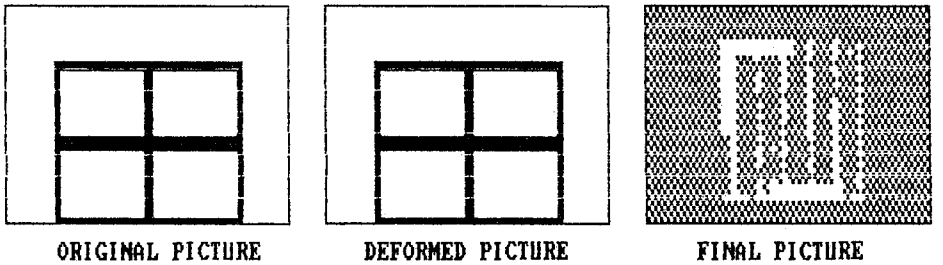
```
DEFORMED AREA                MODE :
NUMBER OF TOGGLED PIXEL  :
NUMBER OF TOTAL PIXEL IS : 3520
DEF
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA % 100          MODE : TOGGLE
NUMBER OF TOGGLED PIXEL  : 1756
NUMBER OF TOTAL PIXEL IS : 3520
DEF % 49.88636
```



ORIGINAL PICTURE DEFORMED PICTURE FINAL PICTURE

PRESS ANY KEY TO CONTINUE.

```
DEFORMED AREA                MODE :
NUMBER OF TOGGLED PIXEL  :
NUMBER OF TOTAL PIXEL IS : 3520
DEF
```