

## BASİT BİR KORELASYON TAHMİNLEYİCİSİ VE BUNUN FRAKTAL GÖRÜNTÜ SIKIŞTIRMADA KULLANIMI

Cengiz GÜNGÖR\*

Aydın ÖZTÜRK\*

### ÖZET

*Doğayla ilgili görüntülerin sıkıştırılmasında diğer yöntemlere göre çok daha etkin olan fraktal görüntü sıkıştırma yöntemi, görüntü içinde kendine benzeyen parçaların bulunması esasına dayanır. Sıkıştırma oranının yüksek tutulduğu durumlarda, fraktal görüntü sıkıştırma yöntemi ile elde edilen görüntülerin kalitesi genellikle diğer yöntemlerden daha iyi olmaktadır. Ancak, benzer parçaların aranması oldukça fazla karşılaştırma hesabı yapılmasını gerektirmektedir. Hesaplama maliyetini düşürmek amacıyla, görüntü üzerinde ele alınan parçaların ve bunlarla eşleştirilmesi öngörülen parçaların sınıflandırılarak, benzerliklerin bu sınıflar içinde aranması en fazla tercih edilen yöntemdir. Bu çalışmada, önerilen sınıflandırma yöntemi ile benzer parçaların basit bir şekilde bulunabileceği gösterilmiştir.*

**Anahtar Kelimeler:** Korelasyon katsayısı, Fraktal görüntü sıkıştırma, Benzerlik ölçüleri

### 1 GİRİŞ

$X$  ve  $Y$  gibi iki değişken arasındaki ilişkinin derecesinin belirlenmesinde aşağıda tanımlanan metrik

$$\rho = \frac{\text{Cov}(X, Y)}{\{\text{Var}(X) \text{Var}(Y)\}^{\frac{1}{2}}} \quad (1)$$

yaygın olarak kullanılmaktadır. Söz konusu değişkenler arasındaki kovaryansın, bunların standart sapmalarına göre standardize edilmesine dayanan ve korelasyon katsayısı olarak bilinen bu ölçü, istatistik literatüründe önemli bir yer tutmuştur.

$X$  ve  $Y$  şans değişkenlerinin geldikleri dağılışın ve bunlardan çekilen örneklerin özelliklerini dikkate alarak, bu değişkenler arasındaki korelasyonu tahmin etmek amacıyla bir çok istatistik geliştirilmiştir. Eldeki problemin özelliğine bağlı olarak, bu istatistiklerinden biri diğerlerine göre daha uygun olabilmekte, birçok durumda ise mevcut şartlar hangi istatistiğin kullanılmasının gerektiğini dikte ettirmektedir.

(\*) Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü, Bornova/İzmir (haberleşme adresi)

Şans örneğine dayanarak değişkenler arasındaki doğrusal ilişkiyi tahmin etmek amacıyla kullanılan Pearson'un korelasyon istatistiği

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\left\{ \sum_{i=1}^n (X_i - \bar{X})^2 \sum_{i=1}^n (Y_i - \bar{Y})^2 \right\}^{\frac{1}{2}}} \quad (2)$$

şeklinindedir. Yukarıdaki eşitlikten de anlaşılacağı gibi, eldeki örnek ve işlenmesi gereken veri hacmi çok büyük olmadığı durumlarda ilgili korelasyon katsayıları kolayca hesaplanabilir. Ancak bazı durumlarda, veri hacmi çok büyük olmakta, özellikle gerçek zamanlı uygulamalarda hesaplama maliyeti çok yüksek olmakta, bu yüzden sonuçların öngörülen zaman aralığında elde edilmesi mümkün olamamaktadır.

(2) deki korelasyon katsayısına ilişkin hesaplama sorunu ile daha önce üzerinde çalıştığımız fraktal görüntü sıkıştırma işlemlerinde ortaya çıkmış ve bu sorunu gidermek amacıyla hesaplama maliyeti daha düşük olan bir korelasyon ölçüsü arayışına girilmiştir. Bilindiği gibi, fraktal görüntü sıkıştırma işlemlerinde, verilen bir görüntü birbiriyle çakışmayan küçük parçalara bölünmekte, bu parçaların benzerleri, bir takım dönüşümlerden sonra yine aynı görüntü üzerinde aranmaktadır (Jacquin, 1990; Fisher, 1995). Genellikle  $m \times m$  ( $m=2, 4, 8, 32, 64$ ) boyutlarındaki kare blokların kullanıldığı bu işlemler sonunda görüntüyü oluşturan birbirleriyle çakışmayan “referans bloklarla” (*range bloklar*), bunların benzerlerini aramak için oluşturulan havuzdaki “test bloklar” (*domain bloklar*), piksel değerleri bazında çok büyük hacimli verileri oluştururlar. Örneğin,  $512 \times 512$  bir görüntüde, işlenmesi gereken blok sayısı yaklaşık 3.200 iken, her blok için yapılması gereken işlem sayısı da yaklaşık 160.000 kadardır. Benzerlik arama işlemlerinde (2)'deki korelasyon katsayısı esas alındığında hesaplama maliyetinin kabul edilebilir sınırların dışına çıkacağı açıkça görülmektedir.

Fraktal görüntü sıkıştırma işlemlerinde birbirine benzeyen blokların bulunmasında izlenen en temel yaklaşım test blokların sınıflandırılması esasına dayanır. Eğer test bloklar belirli bir kritere göre sınıflandırılabilirse o zaman verilen bir referans bloğunun benzeri sadece bu sınıf içinde aranabilir. Bu taktirde problem, eldeki referans bloğunun piksel değerleri ile seçilen sınıf içindeki tüm test bloklarının her birindeki piksel değerleri arasındaki korelasyon katsayılarının hesaplanıp bunların maksimumunun bulunmasına indirgenmiş olur. Bu bağlamda,  $2 \times 2$  boyutlarına indirgenmiş *kanonik* formlar olarak bilinen ve blokların piksel dizilişlerine dayanan çeşitli sınıflandırma şemaları geliştirilmiştir (Boss ve Jacobs, 1991; Jacobs, Boss ve Fisher, 1992; Fisher 1995). Başka bir uygulamada, genetik algoritmalar ile optimize edilen *fuzzy* sınıflandırıcı kullanılmıştır (Loe, Gu ve Phua, 1997). Kominek (1995), çok boyutlu veri uzayını indekslemek için *r-tree* üzerine kurulu bir sınıflandırma şeması geliştirmiş, Saupe (1995), aynı amaçla *kd-tree* kullanmıştır.

Bu çalışmada hesaplanması kolay bir istatistik ileri sürülmüş ve bunun bazı özellikleri üzerinde durulmuştur. Söz konusu istatistiğe dayanan bir sınıflandırma yöntemi geliştirilmiş ve bunun fraktal görüntü sıkıştırmasında uygulaması gösterilmiştir.

## 2 YENİ BİR KORELASYON İSTATİSTİĞİ

$(X_1, Y_1), (X_2, Y_2) \dots (X_n, Y_n)$ , herhangi bir iki-değişkenli (*bivariate*) dağılımdan çekilen bir şans örneği olsun. Bu örnekteki  $i$ 'nci şans değişkeni çifti için ( $i=1, 2, \dots, n$ ), bir şans değişkeni aşağıdaki gibi tanımlanabilir

$$I_i = \begin{cases} 1 & (X_i - \bar{X})(Y_i - \bar{Y}) \geq 0 \text{ veya } (X_i - \bar{X}) = (Y_i - \bar{Y}) = 0 \\ 0 & \text{aksi taktirde} \end{cases} \quad (3)$$

Burada,  $\bar{X} = \sum_{i=1}^n X_i / n$  ve  $\bar{Y} = \sum_{i=1}^n Y_i / n$  dir. Bu çalışmada,

$$Q_n = \frac{1}{n} \sum_{i=1}^n I_i \quad (4)$$

şeklindeki istatistik, korelasyonun bir ölçüsü olarak tanımlanmıştır. Buna göre,  $\{(X_i, Y_i), (i=1,2 \dots n)\}$  şans örneğinde, bir değişken çiftine ilişkin değerlerin ortalamadan ayrılışlarının işaretleri aynı yada ikisinin de değeri sıfır ise,  $Q_n$ 'in hesaplanmasında +1; işaretler farklı ise 0 olarak işlem görmektedir. Böylece, örnekte her iki değişkene ilişkin ortalamadan ayrılışlar hep artan bir seyir izliyorsa  $Q_n=1$ , biri hep artarken diğeri hep azalıyorsa  $Q_n=0$  olacaktır. Diğer taraftan, ortalamadan ayrılışlar birbirinden bağımsız seyrediyorsa bu durumda da  $Q_n$ 'in 0,5'e yakın bir değer alması beklenir. Sonuç olarak  $0 \leq Q_n \leq 1$  dir. Bu özelliği bakımından  $Q_n, r$  ile örtüşmektedir.

Eğer  $X$  ve  $Y$  gibi iki değişken arasındaki ilişki tespit edilmek istenirse, en çok kullanılan istatistik  $r$  istatistiğidir. Fakat uygulamanın anlatıldığı bölümde de açıklanacağı gibi, iki görüntü bloğunun benzerliği test edilirken bloklar büyüdükçe işlemin karmaşıklığı da artmaktadır. En iyi sonuçların alınabilmesi için tüm referans blokları, mevcut tüm test blokları, olası tüm simetrikler de hesaba katılarak test edilmelidir.  $m$  adet referans ve  $n$  adet test bloğu olması durumunda;  $8 \times m \times n$  adet test gereklidir. Örneğin;  $512 \times 512$  ebatlarına sahip bir görüntüde test sayısı 514 milyon civarındadır.

$Q_n$  istatistiği, fraktal görüntü sıkıştırma tekniğinde, en iyi benzerliğin önceden tahmin edilerek, asıl yapılması gereken test sayısını düşürmek için kullanılabilir. Benzer özellikte formlara sahip test blokları benzer listeler altında toplanarak, ilgisiz test bloklarıyla uğraşmadan, istenilen hedefe kolayca ulaşılabilecektir.

### 2.1 $Q_n$ İstatistiğinin Özellikleri

(3) ve (4) deki tanımlardan da anlaşıldığı gibi,  $Q_n$  istatistiği yer ve ölçek (*location* ve *scale*) değişimlerinden bağımsızdır. Bir başka ifadeyle, değişkenler için

yapılacak doğrusal dönüşümler  $Q_n$  istatistiğini etkilemez. Diğer taraftan,  $Q_n$  kesikli bir dağılım gösterir. Büyüklüğü  $n$  olan bir örnekte  $nQ_n$  ifadesi sadece  $\{n, (n-1), \dots, 2, 1, 0\}$  değerlerinden birini alacağından, söz konusu istatistiğin tanımlandığı kümenin eleman sayısı da  $n+1$  olacaktır.  $Q_n$ 'in dağılımı örnek büyüklüğüne ( $n$ ) ve örneğin çekildiği iki değişkenli ana dağılıma bağlıdır. Eğer şans örneği iki değişkenli normal dağılımdan geliyorsa o zaman  $Q_n$ , dağılımın parametresi olan  $\rho$ 'nun bir tahminleyicisi olarak kullanılabilir. Ayrıca, ana dağılıma ilişkin marjinal dağılımların her ikisi de simetrik iseler

$$\lim_{n \rightarrow \infty} \{P(X_i \leq \bar{X})\} = \lim_{n \rightarrow \infty} \{P(Y_i \leq \bar{Y})\} \rightarrow 1/2 \quad (5)$$

olduğundan  $Q_n$ 'in asimptotik dağılımı ana dağılımdan bağımsızdır. Bununla beraber,  $Q_n$ 'in dağılımı, değişkenler arasındaki korelasyona bağlıdır.

(3)'deki ifadede aritmetik ortalamalar yerine medyan kullanıldığı takdirde,  $\rho=0$  olduğuna ilişkin sıfır hipotezi altında  $Q_n$ , ortalaması 0 varyansı  $\sigma^2(n)$  olan normal dağılım gösterir.

$Q_n$ 'in küçük örneklerdeki dağılımını incelemek için Monté Carlo denemeleri yapılmıştır. Bu amaçla, Üniform, Normal ve Üstlü dağılımlardan, ortalaması  $\mu$ , varyansı 1 olan ve aralarında korelasyon bulunmayan  $(U_i, V_i)$ ,  $i=1,2 \dots n$  şeklindeki değişken çiftleri türetilmiştir. Aralarında korelasyon katsayısı  $\rho$  olacak biçimde iki değişkenli dağılım gösteren değişken çiftleri aşağıdaki gibi elde edilmiştir:

$$\begin{aligned} X_i &= U_i \\ Y_i &= \rho U_i + (1 - \rho^2)^{1/2} V_i. \end{aligned} \quad (6)$$

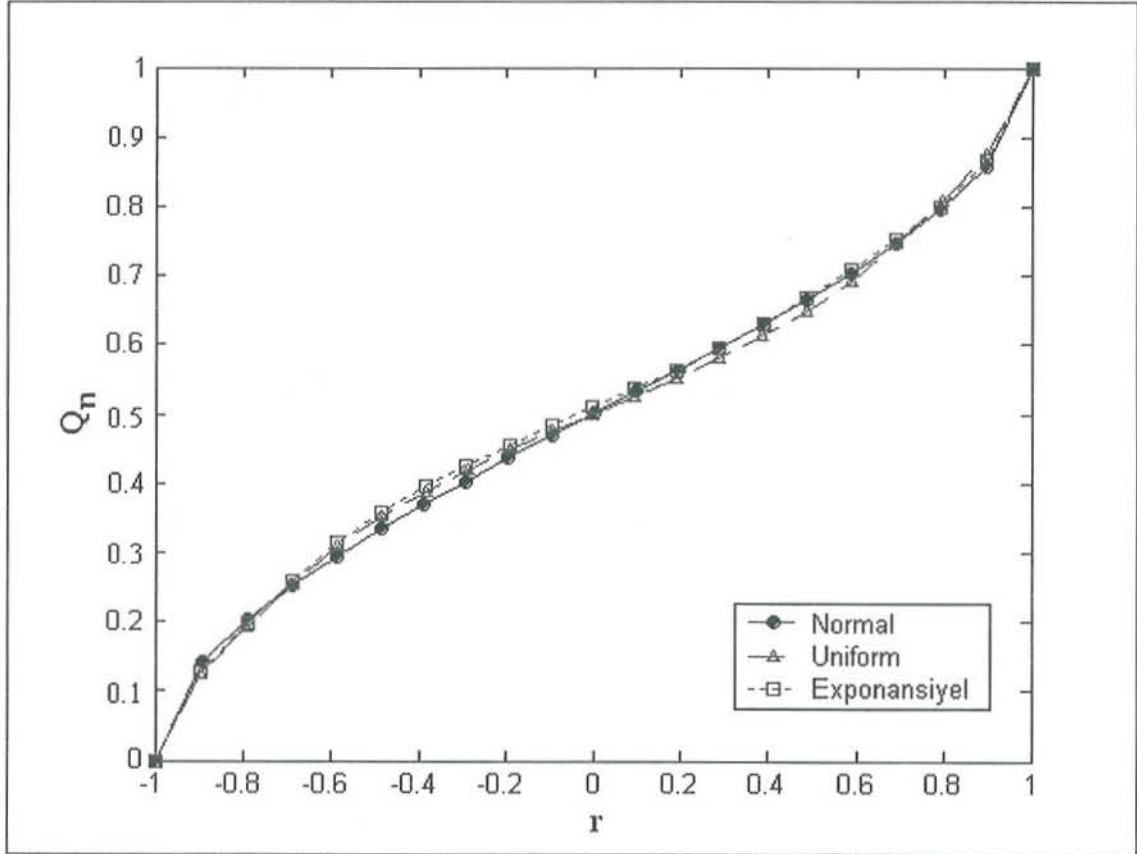
Burada  $\text{Var}(X_i) = \text{Var}(Y_i) = 1$ ,  $\text{Cov}(X_i, Y_i) = \rho$  dur.

$Q_n$  ile  $r$  arasındaki ilişkiyi incelemek üzere Uniform(0, 3.464), Normal(0,1) ve bir parametrelili Üstlü(1) dağılımlardan (6)'daki gibi  $n=16$  büyüklüğünde örnekler türetilmiştir.  $\rho = -1,0$  (0,1) 1,0 değerlerinin her biri için 100.000 örnek türetilmiş ve bunlarla ilgili  $Q_n$  ve  $r$  istatistiklerinin ortalamaları hesaplanmıştır.  $Q_n$  ile  $r$  arasındaki ilişki Şekil 1'de gösterilmiştir.

$Q_n$ 'in örnek dağılımı hakkında fikir edinmek amacıyla çeşitli örnek büyüklükleri seçilerek her bir dağılım için bu istatistiğin ortalama, varyans, çarpıklık (*skewness*) ve basıklık (*kurtosis*) katsayıları hesaplanmıştır.  $n=15, 25, 50, 100, 500, 1000$  büyüklüğündeki örnekler için denemeler  $10^6/n$  defa tekrarlanmış ve elde edilen istatistiklerin ortalamaları hesaplanmış ve sonuçlar Tablo 1, 2 ve 3 de verilmiştir. Bu tablolarda ayrıca  $Q_n$  ile  $r$  arasındaki korelasyon katsayıları verilmiştir. İki korelasyon ölçüsü arasındaki korelasyon katsayılarının, dağılımların üçünde de  $n$  ve  $\rho$  ile ters orantılı olarak değiştiği görülmektedir.

Tablo 1, 2 ve 3 deki ampirik sonuçlardan da görüldüğü gibi,  $\rho=0$  hipotezi altında Normal ve Üniform gibi simetrik dağılımlar için  $Q_n$ 'in dağılımı, ortalaması 0,5 olan

normal dağılıma yaklaşmaktadır. Benzer özelliğin çok çarpık yapıdaki üstlü dağılımda da gözlenmesi ilginç bir sonuç olarak değerlendirilmiştir. Ampirik sonuçlar, örnek büyüklüğü  $n$  ve korelasyon katsayısının  $\rho$ 'nun  $Q_n$ 'in dağılımı üzerindeki etkileri konusunda genel olarak bir fikir vermektedir. Hemen hemen bütün durumlarda basıklık katsayısı 3'e yakın çıkmış, buna karşılık çarpıklık katsayısı  $n$  ve  $\rho$ 'ya bağlı olarak bir miktar varyasyon göstermiştir.



Şekil 1.  $n=16$  ve  $\rho = -1,0 (0,1) 1,0$  için türetilen örneklerden hesaplanan  $Q_n$  ile  $r$  istatistiklerine ilişkin ortalama değerler (Her nokta 100.000 örneğe dayanmaktadır).

**Tablo 1.**  $Q_n$ 'e ilişkin örnek momentleri (Uniform (0, 3,464) Dağılım)

$N$	Ortalama	Varyans	$\rho=0$		
			Çarpıklık	Basıklık	Korelasyon
15	-0,002	0,069	0,011	2,868	0,753
25	0,000	0,041	0,001	2,916	0,751
50	0,001	0,020	0,004	2,964	0,749
100	0,001	0,013	-0,013	2,982	0,746
500	0,000	0,002	-0,005	2,997	0,750
1000	0,000	0,001	0,005	3,003	0,750

$N$	Ortalama	Varyans	$\rho=0,8$		
			Çarpıklık	Basıklık	Korelasyon
15	0,617	0,042	-0,410	3,034	0,526
25	0,617	0,025	-0,342	3,093	0,523
50	0,622	0,012	-0,219	3,012	0,516
100	0,623	0,006	-0,147	3,034	0,503
500	0,625	0,001	-0,049	2,977	0,497
1000	0,624	0,001	-0,024	2,946	0,505

**Tablo 2.**  $Q_n$ 'e ilişkin örnek momentleri (Normal(0,1) Dağılım)

$n$	Ortalama	Varyans	$\rho=0$		
			Çarpıklık	Basıklık	Korelasyon
15	0,000	0,068	0,010	2,870	0,676
25	0,000	0,041	0,009	2,941	0,662
50	0,000	0,020	0,005	2,983	0,654
100	0,000	0,010	0,008	2,961	0,648
500	0,000	0,002	0,000	2,997	0,643
1000	0,000	0,001	-0,003	2,998	0,643

$n$	Ortalama	Varyans	$\rho=0,8$		
			Çarpıklık	Basıklık	Korelasyon
15	0,593	0,044	-0,389	3,052	0,525
25	0,591	0,026	-0,300	3,005	0,506
50	0,590	0,013	-0,254	3,079	0,492
100	0,590	0,007	-0,174	3,031	0,479
500	0,591	0,001	-0,021	2,981	0,488
1000	0,591	0,001	-0,077	3,012	0,471

**Tablo 3.**  $Q_n$ 'e ilişkin örnek momentleri (Üstlü(1) Dağılım)

$n$	Ortalama	Varyans	$\rho=0$		
			Çarpıklık	Basıklık	Korelasyon
15	0,020	0,068	0,034	2,787	0,734
25	0,025	0,039	-0,004	2,932	0,723
50	0,021	0,020	0,020	2,907	0,719
100	0,021	0,010	0,024	2,999	0,708
500	0,022	0,002	-0,022	3,048	0,709
1000	0,022	0,001	-0,023	3,037	0,710

$n$	Ortalama	Varyans	$\rho=0,8$		
			Çarpıklık	Basıklık	Korelasyon
15	0,604	0,044	-0,403	2,975	0,585
25	0,601	0,027	-0,353	3,140	0,585
50	0,605	0,013	-0,277	3,054	0,588
100	0,601	0,007	-0,191	3,024	0,592
500	0,602	0,001	-0,047	3,005	0,592
1000	0,603	0,001	-0,061	2,981	0,594

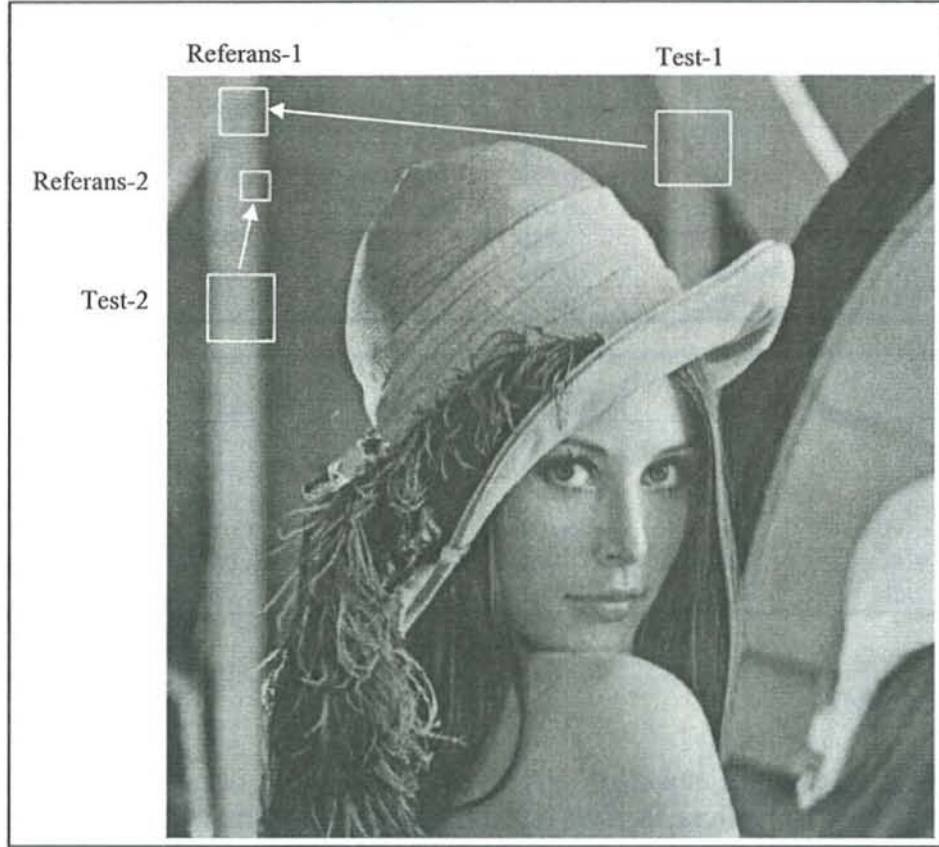
### 3 UYGULAMA

Doğal şekillerin noktalar, çizgiler ve düzlemler yardımıyla görüntülenmesi oldukça zor bir iştir. Oysa ilk defa 1980'li yıllarda Mandelbrot'un geliştirdiği fraktal geometri teknikleri ile dağ, bulut, bitki örtüsü gibi doğal şekillerin kolayca oluşturulması mümkün olmuştur. Fraktal geometri, temelde bir başlangıç resmine belli bir dönüşümün tekrar tekrar uygulanarak sonuçta sabitleşen bir resminin elde edilmesine dayanmaktadır. Bir başka ifadeyle doğal şekiller bir takım dönüşümleri temsil eden matrislerle oluşturulabilmektedir. Bu tür dönüşümler yardımıyla söz konusu objeler oluşturup birleştirilerek bir doğa manzarası elde edilebilmektedir (Barnsley, 1992).

Fraktal geometri uygulamalarından elde edilen olumlu sonuçlardan sonra aklı gelen ilk soru işlemin tersine işleyip işlemeyeceği olmuştur. Bu noktadan hareket eden matematikçiler, bir resmin içinden alınacak parçalarla o resmin bir benzerini bir takım dönüşümlerden sonra elde etmenin mümkün olabileceğini göstermişlerdir. Bu konudaki ilk çalışma Barnsley (1988) tarafından yapılmıştır. Jacquin (1992) eldeki resmi parçalara ayırıp her parça için ayrı bir dönüşüm bulup bunları birleştirerek orijinal resmin benzerini elde eden bir algoritma geliştirmiştir. Bu yöntemin en önemli uygulaması görüntü sıkıştırma konusunda olmuştur.

#### 3.1 Fraktal Görüntü Sıkıştırma

Fraktal görüntü sıkıştırma tekniği, görüntü üzerinde oluşturulan referans blokları ile, yine aynı görüntü üzerinde referans bloklardan daha büyük olacak şekilde seçilen test bloklarının eşleştirilmesi esasına dayanır. Referans blokları birbirleriyle çakışmayacak şekilde düzenlendikleri halde test blokları için bu kural geçerli değildir. Referans ve test bloklara birkaç örnek Şekil 2'de gösterilmiştir.



Şekil 2. Lena görüntüsü üzerinde referans ve test blokların eşleştirilmesi

Genel olarak test blokların büyüklüğü referans blokların iki misli olarak seçilir. Ancak birbirine bitişik kare şeklindeki alanda 4 hücrenin ortalaması alınarak elde edilen küçültülmüş boyutlardaki bloklar referans blokları ile eşleştirilir. Bu bloklarda yer alan piksel değerleri arasında basit doğrusal ilişkinin olduğu varsayılırsa bunlar için

$$r_i = \alpha + \beta D_i + e_i \quad i = 1, 2, \dots, n \quad (7)$$

şeklinde basit *regresyon* modeli yazılabilir. Burada  $r_i$  ve  $D_i$  sırasıyla referans ve test bloklara ilişkin piksel değerlerini,  $\alpha$  ve  $\beta$  sırasıyla parlaklık ve kontrastla ilgili parametreleri  $e_i$  ise hata terimini göstermektedir. Bu modelin parametreleri en küçük kareler yöntemine göre tahmin edildikten sonra RMS(*Root Mean Square*) değeri aşağıdaki gibi hesaplanabilir:

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n (r_i - \hat{\alpha} - \hat{\beta} D_i)^2} \quad (8)$$

Burada  $\hat{\alpha}$  ve  $\hat{\beta}$  parametre tahminleridir. Buna göre eldeki bir referans blok, havuzdaki tüm test bloklarla eşleştirilerek bu çiftler için RMS değerleri hesaplanır. RMS değeri en küçük olan durumdaki test blok eldeki referans bloğa en çok benzeyen blok olarak



seçilir. Bu seçimde pozitif korelasyon esas alındığından, minimum RMS değerini veren eşleştirmeler yapılırken bu durumun dikkate alınması gerekir.

Resim üzerindeki bloklar her zaman birbirlerine tam olarak benzemezler. Aralarında kontrast ve parlaklık farkı ile piksel dizilişlerinde farklılıklar olabilir. Piksellerin diziliş farkı 8 simetri işlemi ile ifade edilebilir. Bunlar; bloklar resimden alındıkları hali ile benzeyebilirler veya 90°, 180° ve 270° çevrilmiş halleri ile benzeyebilirler, diğer dört diziliş ise ilk dört işlemin  $x$ -eksenine göre simetrik halleridir. Bloklar arasındaki kontrast ve parlaklık farkı ile 8 simetri işleminden hangisinin uygulanacağı kolayca bulunabilmektedir (Fisher, 1995).

256 gri seviyeli iki resim arasındaki kalite farkı ise aralarındaki RMS ölçümünün bir fonksiyonu olan PSNR (*Peak Signal to Noise Ratio*) ile ifade edilir :

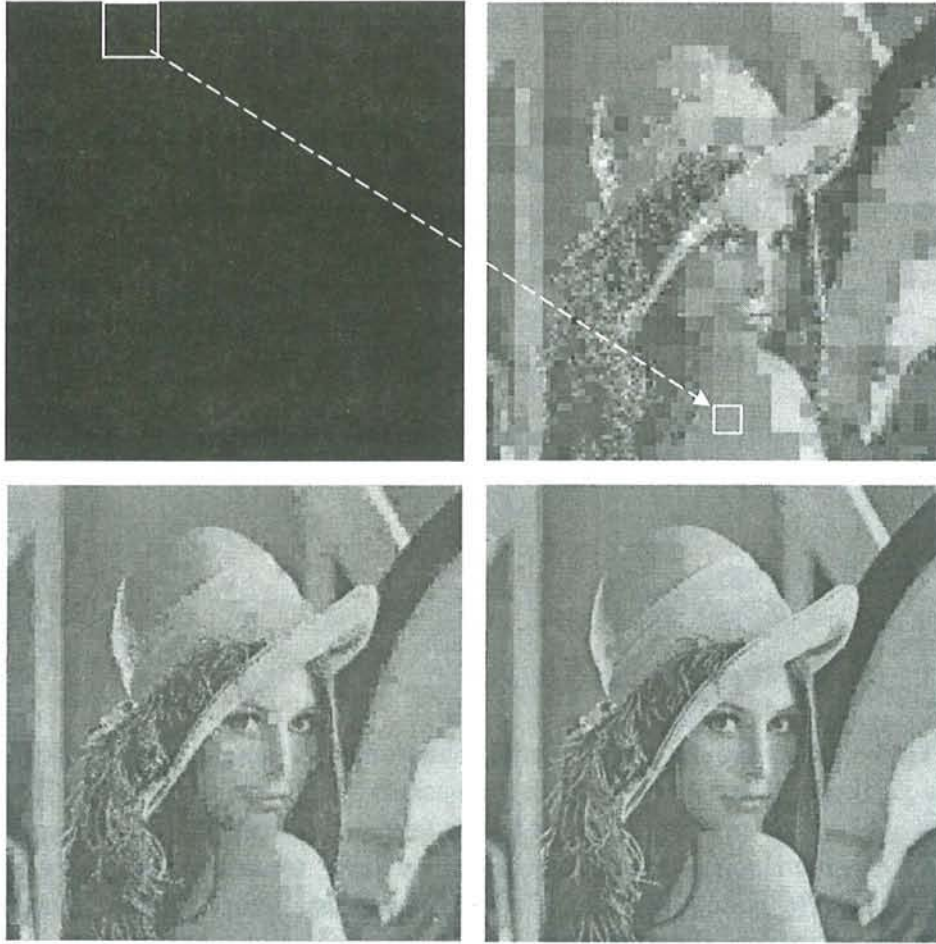
$$PSNR = 20 \log_2 \left( \frac{255}{RMS} \right) \quad (9)$$

### 3.2 Resmin Oluşturulması (*Decoding*)

Tüm referans bloklarının kodlanmış halleri resmin tamamını ifade eden bir dosya içerisine yazılırlar. Dosyadan geri açma aşamasında Şekil 3’de gösterildiği gibi siyah bir resimden başlanarak tüm dönüşümler tekrar tekrar uygulanarak sonuç resme ulaşılır. Belli bir tekrardan sonra işleme devam edilse dahi sonucun değişmediği gözlenir. Resmin oluşturulmasında çok fazla işlem karmaşası olmadığından sonuca birkaç tekrarla hızlı bir şekilde ulaşılır (Şekil 3).

Resmi kodlarken sabit boyutlu referans bloğu kullanmak sabit sıkıştırma oranı döndürmektedir. Bunun yerine, referans blokların boyutlarını  $2^n$  olacak şekilde birkaç farklı boyutta seçip, büyük boyutlu referans bloklardan başlayarak eşleştirme yapmak daha etkin bir yaklaşım olarak benimsenmiştir. Eldeki referans bloğa uygun bir eş bulunamaması durumunda blok dörde bölünüp, küçük parçaların benzerlerinin aranması yoluna gidilir. Bu yaklaşım literatürde *quadtree* yöntemi olarak bilinir (Fisher, 1995). Bu teknikte çok basit bir şekilde, iyi bir sıkıştırma elde edilebilmektedir.

Fraktal görüntü sıkıştırma konusunda, arama işlemini hızlandırmak amacıyla farklı sınıflandırma teknikleri geliştirilmiştir. Bir görüntüden elde edilen referans ve test sayıları sırasıyla  $k$  ve  $l$  ise herhangi bir sınıflandırma yapmadan tüm eşleştirme yapma işlemlerinin karmaşıklığı  $O(kl)$  dir. Hesaplama maliyetini düşürmek için hem *quadtree* hem de sınıflandırma stratejilerinin birlikte uygulanması gerekmektedir.



Şekil 3. Kodlanmış Lena görüntüsünün siyah bir resimden geri açılması adımları. Üstte başlangıç resmi (tümüyle siyah) ve birinci aşama, altta ise ikinci ve üçüncü tekrarlar görülmektedir (sıkıştırma oranı 20:1 ve PSNR 32,7).

### 3.3 $Q_n$ İstatistiğinin Uygulanışı

Referans-test eşleştirme işleminden önce test bloklarını basitçe sıralamak için  $Q_n$  istatistiğinde kullanılan tekniğin bir benzeri sınıflandırma için kullanılabilir. Daha sonra sınıflandırılmış listeler içerisinde arama yapılır. Bunun için izlenen aşamalar şu şekildedir.

- a.  $4 \times 4$  'dan büyük bloklar, komşu piksellerin ortalaması alınarak  $4 \times 4$  boyutlarına indirgenir. İndirgenmiş bloktaki  $i$ 'nci piksel değeri için tanımlanan

$$J_i = \begin{cases} 1 & \text{eğer } (x_i \geq \bar{x}) \text{ ise} \\ 0 & \text{aksi durumda} \end{cases} \quad i = 1, 2, \dots, 16 \quad (10)$$

kullanılarak

$$k = \sum_{i=1}^{16} J_i \quad (11)$$

şeklinde bir değer bulunur. Tüm test bloklar için  $k$  değeri hesaplanarak, test bloklar bu değere göre sınıflandırılır. Bu duruma göre en çok 16 sınıf oluşturulabilir. test bloklar sınıf numaraları aracılığı ile, 16 elamanlı listede ilgili numaralı alt listeye eklenirler.

- b. Ele alınan bir referans blokla ilgili sınıf numarası da (11)'deki gibi bulunur.
- c. Referans bloğunun benzeri, test bloklarından oluşan havuzda, aynı sınıf numarasındaki bloklar içinde aranır. Eşleştirme işlemi hızlandırmak için  $Q_n$  istatistiği kullanılır. Bu amaçla, her iki bloğun  $J_i$  değerleri 16 bitlik tamsayı değişkenlerinde saklanırsa, bu değişkenler XOR'lanarak ve elde edilen sonuçtaki '1' olan bitler sayılarak,  $Q_n$  basitçe hesaplanabilir (Mano, 1993). Bu aşamada  $Q_n$  için belirlenen bir eşik değerine göre (bu çalışmada eşik  $Q_n=0,75$  seçilmiştir) test bloklar ayıklanarak geri kalanlar arasında minimum RMS değerini veren referans-test çifti aranır.
- d. Listedeki tüm test bloklarla karşılaştırma tamamlandığında bulunan minimum RMS değeri önceden belirlenen eşik değerinden daha büyük çıkarsa, komşu bir alt sınıf, bir üst sınıf ( $\Delta=1$ ), iki alt sınıf, iki üst sınıf ( $\Delta=2$ ), üç alt sınıf, üç üst sınıflardaki ( $\Delta=3$ ) test bloklarında benzerlik bulununcaya kadar arama işlemlerine devam edilir.
- e. Yukarıdaki son adımda da benzerlik bulunamamışsa o zaman eldeki referans blok dört eşit parçaya bölünerek arama işlemleri her bir alt blok için öz-yineli (*recursive*) olarak tekrarlanır.

$\Delta=3$  sınırı çeşitli görüntüler üzerinde yapılan denemelerden sonra kabul edilebilecek optimum değer olarak belirlenmiştir. Lena görüntüsü için sınıflara göre test bloklarının dağılımı Tablo 4'de verilmiştir (diğer görüntülerde de benzer bir dağılım vardır). Tablo 5, 6 ve 7'de  $\Delta$  değerlerine göre eşleştirme dağılımları verilmiştir. Tablo 8, 9 ve 10'da ise  $\Delta$  değerinin sıkıştırma oranı, PSNR ve toplam süre üzerine etkileri verilmiştir. Ayrıca aynı tablolarda *Brute-force* ve Saupe'un *kd-tree* sınıflandırma yöntemi ile elde edilen değerler ile karşılaştırmalar yapılmıştır (Değerler 1GB bellekli, Pentium IV 3Ghz makine ile alınmıştır).

**Tablo 4.** 512x512 Lena test görüntüsünde, farklı boyutlarda, sınıf listeleri altındaki test blokları sayılarının dağılımı.

Blok Boyutu	$k$															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
4x4	1	17	63	294	624	1586	3007	4961	3114	1605	586	193	53	8	0	0
8x8	0	23	141	493	691	1469	2559	4393	2792	1646	765	544	83	24	2	0
16x16	8	48	200	790	766	1465	2219	3570	2218	1414	884	862	164	28	5	0
32x32	0	6	77	671	816	1535	2160	3130	1931	1274	724	306	125	14	0	0

**Tablo 5.** Lena görüntüsünde en iyi referans-test eşleştirmelerinin  $\Delta$  değerlerine göre dağılımı

Blok Boyutu	$\Delta=0$	$\Delta=1$	$\Delta=2$	$\Delta=3$	Toplam
4x4	1.834 (%84,3)	213 (%9,8)	87 (%4,0)	42 (%1,9)	2.176
8x8	649 (%92,7)	43 (%6,1)	8 (%1,1)	0 (%0,0)	700
16x16	265 (%90,4)	22 (%7,5)	6 (%2,0)	0 (%0,0)	293
32x32	98 (%93,3)	7 (%6,7)	0 (%0,0)	0 (%0,0)	105

**Tablo 6.** Mandrill görüntüsünde en iyi referans-test eşleştirmelerinin  $\Delta$  değerlerine göre dağılımı

Blok Boyutu	$\Delta=0$	$\Delta=1$	$\Delta=2$	$\Delta=3$	Toplam
4x4	4.867 (%40,0)	2.672 (%22,0)	2.537 (%20,9)	2.060 (%17,0)	12.136
8x8	508 (%90,4)	43 (%7,7)	7 (%1,2)	4 (%0,7)	562
16x16	96 (%91,4)	6 (%5,7)	0 (%0,0)	3 (%2,9)	105
32x32	5 (%100,0)	0 (%0,0)	0 (%0,0)	0 (%0,0)	5

**Tablo 7.** Goldhill görüntüsünde en iyi referans-test eşleştirmelerinin  $\Delta$  değerlerine göre dağılımı

Blok Boyutu	$\Delta=0$	$\Delta=1$	$\Delta=2$	$\Delta=3$	Toplam
4x4	4.167 (%81,7)	510 (%10,0)	257 (%0,5)	166 (%3,3)	5.100
8x8	1.220 (%90,7)	103 (%7,7)	12 (%0,8)	10 (%0,7)	1.345
16x16	210 (%93,3)	11 (%4,9)	2 (%0,1)	2 (%2,9)	225
32x32	35 (%97,2)	1 (%2,8)	0 (%0,0)	0 (%0,0)	36

**Tablo 8.** Lena görüntüsünde.farklı  $\Delta$  değerleri ile elde edilen sonuçlar.

Kullanılan Yöntem	Sıkıştırma Oranı	PSNR	Test Sayısı	Süre (sn)
Önerilen ( $\Delta = 0$ )	20,67 : 1	33,771	2.966.642	3,34
Önerilen ( $\Delta \leq 1$ )	22,08 : 1	33,523	4.817.309	5,47
Önerilen ( $\Delta \leq 2$ )	22,36 : 1	33,497	5.541.897	6,61
Önerilen ( $\Delta \leq 3$ )	22,54 : 1	33,454	6.271.008	7,56
Brute Force	22,72 : 1	33,743	65.508.848	34,53
Saupe kd-tree	22,11 : 1	33,749	216,400	1,39

**Tablo 9.** Mandrill görüntüsünde.farklı  $\Delta$  değerleri ile elde edilen sonuçlar

Kullanılan Yöntem	Sıkıştırma Oranı	PSNR	Test Sayısı	Süre (sn)
Önerilen ( $\Delta = 0$ )	5,68 : 1	26,747	3.781.774	8,53
Önerilen ( $\Delta \leq 1$ )	5,75 : 1	27,382	13.253.637	21,50
Önerilen ( $\Delta \leq 2$ )	5,76 : 1	27,483	15.501.771	28,00
Önerilen ( $\Delta \leq 3$ )	5,77 : 1	27,513	18.132.796	32,44
Brute Force	5,79 : 1	27,663	268.937.796	105,41
Saupe kd-tree	5,76 : 1	27,527	850,600	5,84

**Tablo 10.** Goldhill görüntüsünde farklı  $\Delta$  değerleri ile elde edilen sonuçlar

Kullanılan Yöntem	Sıkıştırma Oranı	PSNR	Test Sayısı	Süre (sn)
Önerilen ( $\Delta = 0$ )	10,30 : 1	32,154	4.521.444	5,61
Önerilen ( $\Delta \leq 1$ )	10,82 : 1	32,179	8.412.712	9,86
Önerilen ( $\Delta \leq 2$ )	10,89 : 1	32,182	9.819.364	12,00
Önerilen ( $\Delta \leq 3$ )	10,95 : 1	32,169	11.315.240	13,80
Brute Force	11,06 : 1	32,489	137.880.412	61,48
Saupe kd-tree	10,71 : 1	32,484	452.600	3,13

Tablolarda üç farklı görüntü için elde edilen sonuçlardan da anlaşılacağı üzere sınıflandırma şeması ve  $Q_n$  istatistiği test edilen blok çiftlerinin sayısını 70 kata kadar azaltmış, bu da “brute-force” karşısında zaman olarak 12 kata kadar hız sağlamıştır. Ayrıca  $\Delta \leq 1$  iken tatmin edici sonuçlara ulaşıldığı görülmektedir (Şekil 4).



Şekil 4. Solda orijinal Lena, ortada  $\Delta \leq 1$  için ve sağda  $\Delta \leq 3$  için elde edilen görüntüler



Şekil 5. Solda orijinal Mandrill, ortada  $\Delta \leq 1$  için ve sağda  $\Delta \leq 3$  için elde edilen görüntüler



Şekil 6. Solda orijinal Goldhill, ortada  $\Delta \leq 1$  için ve sağda  $\Delta \leq 3$  için elde edilen görüntüler

### KAYNAKLAR

- BARNESLEY, M. (1992), *Fractals Everywhere*, Academic Press, San Diego, CA, USA.
- BOSS, R.D. and JACOBS, E.W. (1991), *Studies of iterated transform image compression, and its application to color and DTED.*, Technical report 1468, Naval Ocean Systems Center, San Diego, CA.
- FISHER, Y. (1995), *Fractal image compression, Theory and application.*, Springer-Verlag.
- JACOBS, E.W., BOSS, R.D. and FISHER, Y. (1992), *Image compression: A study of the iterated transform method*, Signal Processing, 29:251-263.
- JACQUIN, A. (1990), *A novel Fractal block coding technique for digital images*, IEEE ICASP Proc., 4:2225-2228.
- KOMINEK, J. (1995), *Advances in Fractal Compression for Multimedia Applications*, To appear in Multimedia System Journal.
- MANDELBROT, B. (1983), *The Fractal Geometry of Nature*, W.H.Freeman & Co., Second Edition.
- LOE, K.F., GU, W.G. and PHUA, K.H. (1997), *Speed-up fractal image compression with a fuzzy classifier*, Signal Processing: Image Communication 10: 303-311.
- MANO, M.M. (1993), *Computer System Architecture*, Prentice Hall International Inc., pp 6-7.
- SAUPE, D. (1995), *Accelerating fractal image compression by multi-dimensional nearest neighbor search*. In J. A. Storer and M. Cohn, editors, *Proceedings DCC'95 (IEEE Data Compression Conference)*, pages 222-231, Snowbird, UT, USA, March 1995.
- SOLOMON, D. (2000), *Data compression. The complete reference*. Springer-Verlag Inc. 240-242

## A SIMPLE ESTIMATOR OF CORRELATION AND ITS APPLICATION ON FRACTAL IMAGE COMPRESSION

### ABSTRACT

*Fractal image compression which is based on finding similar image blocks, in an efficient method for compression of similar pieces. Fractal image compression method generally obtains more realistic results than the other methods on high compression rates. However, searching similar pieces requires tedious computations. In order to reduce the computational coast domain blocks are classified and search for finding the similar pairs are performed within those classes. In this study, it is shown that similar pairs can be obtained by proposed classification method.*

**Key Words:** *Correlation coefficient, Fractal image compression, Similarity measures.*