

2008-58

TÜBİTAK

TÜRKİYE BİLİMSEL VE TEKNOLOJİK ARAŞTIRMA KURUMU
THE SCIENTIFIC AND TECHNOLOGICAL RESEARCH COUNCIL OF TURKEY

Elektrik, Elektronik ve Enformatik Araştırma Grubu
Electrical, Electronical and Informatics Research Group

92645

**Üç Boyutlu Göz Geometrisinin Küme Bölüntüleme
ile Verimli Spektral Kodlaması**

Proje No: 106E064

Doç.Dr. Uluğ Bayazıt
Yrd. Doç. Dr. Hasan Fehmi Ateş
Umut Konur

Ağustos 2008
İSTANBUL

Önsöz

İçindekiler

106E064 nolu TÜBİTAK projesi Işık Üniversitesi ve İstanbul Teknik Üniversitesi bünyesinde Eylül 2006-Haziran 2008 tarihleri arasında yürütülmüştür.

Çalışma 01

Çalışma 02

Çalışma 03

Çalışma 04

Çalışma 05

Çalışma 06

Çalışma 07

Çalışma 08

Çalışma 09

Çalışma 10

Çalışma 11

Çalışma 12

Çalışma 13

Çalışma 14

Çalışma 15

Çalışma 16

Çalışma 17

Çalışma 18

Çalışma 19

Çalışma 20

Çalışma 21

Çalışma 22

Çalışma 23

Çalışma 24

Çalışma 25

Çalışma 26

Çalışma 27

Çalışma 28

Çalışma 29

Çalışma 30

Çalışma 31

Çalışma 32

Çalışma 33

Çalışma 34

Çalışma 35

Çalışma 36

Çalışma 37

Çalışma 38

Çalışma 39

Çalışma 40

Çalışma 41

Çalışma 42

Çalışma 43

Çalışma 44

Çalışma 45

Çalışma 46

Çalışma 47

Çalışma 48

Çalışma 49

Çalışma 50

Çalışma 51

Çalışma 52

Çalışma 53

Çalışma 54

Çalışma 55

Çalışma 56

Çalışma 57

Çalışma 58

Çalışma 59

Çalışma 60

Çalışma 61

Çalışma 62

Çalışma 63

İçindekiler:

Önsöz.....	2
İçindekiler.....	3
Tablo ve Şekil Listeleri	4
Özet	5
Abstract.....	6
1. Giriş	7
2. Genel Bilgiler.....	12
2.1 Tel file yüzey bölüntüleme	12
2.2 Spektral dönüşüm.....	12
2.3 Spektral tel file sıkıştırma yöntemi	14
2.4 Değişmez taban vektörleri kullanılarak düzensiz tel file geometrilerinin kodlanması	15
3. Gereç ve yöntem.....	19
3.1 CSPECK algoritmasının tel file geometri sıkıştırmaya uygulanması.....	19
3.2 CSPECK sembollerinin aritmetik kodlanması.....	22
3.3 Geliştirilen ikili ve aritmetik kodlayıcılar için kod çözücülerin gerçekleştirilmesi.....	23
3.4 Basit bir örnek.....	23
3.5 Önemlilik testi karar sembollerinin birleşik ve basit koşullu kodlanması... ..	27
3.6 Önemlilik testi karar sembollerinin koşullu kodlanması ve uygun bağlamların araştırılması.	28
3.7 Değişmez taban vektörleri ile verimli geometri kodlama için çoktan teke eşlemler.....	31
4. Bulgular ve Tartışma	36
4.1 Ölçevler	37
4.2 Uyarlamalı dönüşüm yönteminin kullanıldığı önerilen CSPECK kodlayıcı ile elde edilen sonuçlar.....	38
4.3 Önerilen değişmez taban vektörler ile dönüşüm yöntemi ile elde edilen sonuçlar.....	42
4.4 Hesaplama karmaşıklığı ölçümleri	45
5. Sonuç.....	46
Yararlanılan kaynaklar.....	47

Tablo ve Şekil Listeleri

a) Tablolar

Tablo 1 x, y ve z için katsayılar satır vektörleri	25
Tablo 2 Her kodlama geçişinde işlenen bir düzlemini gösteren n değeri, katsayı büyüklüklerinin önemini sınamada kullanılan eşik değeri T_n , katsayıların önemlilik durumları, o geçişte önemli bulunan katsayıların kodlanan işaret sembolleri ve o geçiş sonrasında katsayıların geriçatılan büyüklükleri	26
Tablo 3 x,y ve z koordinatları için katsayıların geriçatımları	27
Tablo 4 Bunny modeli için uyarlamalı dönüşüm yönteminin kullanıldığı önerilen CSPECK kodlayıcı (bağlama koşullu sembol kodlama yöntemi) ve dalgacık tabanlı kodlayıcı (KHODAKOVSKY, 2002) için Hausdorff (kuşatan kutu diyagonaline göre) ve etkin değer uzaklıklarının hıza göre değişimi	39

b) Şekiller

Şekil 1 Önerilen dönüşüm tabanlı geometri sıkıştırma yönteminin şematik gösterimi	9
Şekil 2 MeTiS sıradüzensel bölüntüleme yazılımı ile 40 bölgeye ayrılan Horse modelinin bölgelere göre renklendirilen yüzey kaplaması	12
Şekil 3 Özdeğerler küçükten büyüğe doğru sıralandığında karşılık gelen katsayıların enerjilerinin hızla azaldığı gözlemlenmektedir.	14
Şekil 4 4x4'lük düzenli altıgen tel file bölgesi	16
Şekil 5 Birim dairedeki örnek nokta kümesinin altkümelere bölüntülenmesi. Her özyineleme adımında noktalar yaklaşık eşit büyüklükteki iki kümeye bölüntülenmektedir. Her alt bölüntülemeyi gösteren çizgilerin kalınlıkları azalmaktadır. En alt seviyedeki bölüntüleme kesikli çizgiler ile gösterilmektedir. En alt seviyedeki her altküme iki veya üç nokta içermektedir.	17
Şekil 6 <i>S ve I Tipi Kümelerin Bölüntülenmesi</i>	20
Şekil 7 VRML dilinde bir tel file örneği	24
Şekil 8 Örnek tel file geometrisinin geriçatımının VRML dilindeki dosyaya kaydedilmesi	27
Şekil 9 Dönüşüm öncesinde düzensiz tel file bölgesini temsil eden birim dairedeki noktaların (sağda) düzenli tel file bölgesini temsil eden birim dairedeki noktalara (solda) çoktan teke eşlemlenmesi. Bu örnekte $N=5$ olup $K=3$ alınmıştır. Sağdaki birim dairedeki üç noktanın soldaki birim dairede eşlemlendiği noktanın konumu sağdaki birim daire üzerinde açık renkle belirtilmiştir.	32
Şekil 10 Uyarlamalı dönüşüm yönteminin kullanıldığı önerilen CSPECK kodlayıcının sınanması için benzetim sonuçları a) Bunny, b) Horse ve c) Venus tel fileleri	41
Şekil 11 Horse modelinin uyarlamalı dönüşüm yönteminin kullanıldığı önerilen CSPECK kodlayıcı ile aşamalı geriçatımları (a)0.22 bpv, (b)0.81 bpv, (c)7.12 bpv, (d)14.34 bpv	41
Şekil 12 Bunny modeli a): Özgün model, b) ve c): 4.67 bit/düğüm bit hızında sırasıyla dik tel file sıkıştırma yöntemi (KHODAKOVSKY, 2002) ve spektral sıkıştırma yöntemi (KARNI, 2000) ile geriçatımlar d): 4.51bit/düğüm bit hızında uyarlamalı dönüşüm yönteminin kullanıldığı önerilen CSPECK kodlayıcı ile geriçatım	42
Şekil 13 Önerilen değişmez taban vektörler ile dönüşüm yönteminin sınanması için yapılan benzetimlerin sonuçları: a) Bunny, b) Horse ve c) Venus tel fileleri. 44	

Özet

3-B nesnelerin görselleştirilmesi uygulamalarında kullanılan tel filelerin (gözlerin) geometrilerinin sıkıştırılması (DEERING, 1995)'deki öngörücü kodlamaya dayanan yöntem ile beraber rağbet gören bir araştırma alanı olmuştur. Yakın geçmişte araştırmacıların ilgi odağı (KARNI, 2000) ve (KHODAKOVSKY, 2002)'dekine benzer dönüşüm tabanlı tel file geometri sıkıştırma yöntemlerine kaymıştır.

(KARNI, 2000)'de klasik Fourier dönüşümünün 3 boyutlu tel file geometrilerine genelleştirilmesi ile elde edilen dönüşümün düzenli olmayan tel file geometrilerine uygulanması ile özellikle düşük bit hızlarında yüksek sıkıştırma başarımı sağlanmıştır. Bu projedeki çalışmanın ilk aşamasında literatürdeki yüksek başarımlı sağlayan dalgacık dönüşüm tabanlı görüntü kodlama yöntemlerinde (ISLAM, 1999) (PEARLMAN, 2004) dalgacık dönüşüm katsayılarının verimli sıkıştırılması için kullanılmış olan sıradüzensel küme bölüntüleme tekniği, (KARNI, 2000)'de önerilen spektral dönüşüm katsayılarının bütün bit hızlarında verimli sıkıştırılması için uygulanmıştır.

Kodlanan düzensiz tel filenin spektral taban vektörlerinden oluşturulan dönüşümün hesaplama karmaşıklığı çok yüksek olduğu için (KARNI, 2001)'de önerildiği gibi düzenli bir tel filenin değişmez spektral taban vektörleri ile sağlanan dönüşümün düzensiz tel filenin geometri kodlamasında kullanılması da incelenmiştir. Bu dönüşüm yönteminin uygulanmasında kodlanan düzensiz ve taban vektörleri kullanılan düzenli tel fileler arasındaki birebir eşlemlenin eniyilikten uzak olduğu ve dolayısıyla sıkıştırma başarımı için yetersiz kaldığı gözlemlenmiştir. Düzenli ve düzensiz göz geometrileri arasında çoktan bire eşlemlerinin kullanılmasını yüksek bit hızlarında dönüşüm tabanlı kodlamadan uzamsal kodlamaya geçilmesi ile birleştiren önerilen alternatif yöntemin çok daha verimli olduğu gösterilmiştir.

Abstract

Mesh geometry compression, used in the applications related to the visualization of 3-D objects, has become a popular research area following the pioneering work of (DEERING, 1995) that employs simple predictive coding. In the recent past, the mesh geometry compression methods of (KARNI, 2000) and (KHODAKOVSKY, 2002), regarded by most as the state of the art, are transform based.

In (KARNI, 2000), the transform obtained by a generalization of the classical Fourier transform to the 3-D mesh geometries has been applied to irregular meshes to yield a high compression performance especially at low rates. In the first stage of the work of this project, a hierarchical set partitioning technique proposed for the efficient compression of wavelet transform coefficients in high performance wavelet based image coding methods (ISLAM, 1999) (PEARLMAN, 2004) in the literature has been applied to efficiently compress the coefficients of the spectral transform proposed in (KARNI, 2000).

Since the computational complexity of the transform constructed by the spectral bases of the coded irregular mesh is overwhelming, the transform proposed in (KARNI, 2001) that is constructed by the fixed spectral basis of regular meshes has been investigated for use in geometry coding. In the implementation of this transformation method, the one-to-one mapping between the coded irregular mesh and the regular mesh whose basis are being used, has been observed to be rather suboptimal and yield insufficient compression performance. The proposed alternative method that combines the use of many-to-one maps between the regular and irregular mesh geometries and the transition from transform based coding to spatial coding at high rates has been shown to be more efficient.

1. Giriş

Görselleştirme amacıyla 3 boyutlu nesne modellerine iletişim ağları üzerinden erişim özellikle son on yılda birçok uygulamanın temelini oluşturmuştur. Büyük 3 boyutlu nesne model bilgileri iletilirken iletişim bant genişliğinin verimli kullanılması gerekmektedir. Bu gereksinim 3 boyutlu nesne yüzey bilgilerinin kodlanmasına yönelik çalışmaları ön plana çıkarmıştır. Günümüzde birçok sıkıştırma algoritması tel filelerle gösterilen nesnelere üzerinde çalışmakta, VRML, MPEG-4 gibi dosya formatları ve gösterim ve kodlama standartları tel fileler için sıkıştırma araçları içermektedir.

Tel file kodlama esnasında hem topoloji bilgisinin hem de noktaların koordinat değerlerinin kodlanması gerekmektedir. Bağlanırlık kodlama olarak da adlandırılan topolojik bilginin kodlanması alanında yapılan çalışmalar (ROSSIGNAC, 1999) (GUMHOLD, 1998) (ISENBURG, 2000) (TAUBIN&ROSSIGNAC, 1998) (TOUMA, 1998) önceleri daha fazla önemsenmişse de tel file geometri bilgisinin toplam bilginin çok daha büyük bir bölümünü oluşturması nedeniyle öngörücü kodlama (DEERING, 1995) ile başlayan tel file geometri sıkıştırma çalışmaları yakın geçmişte araştırma ilgi odağı olmuştur.

Tel file geometrilerinin kodlamasını gerçekleyen birçok yöntem, noktaların 3 boyutlu koordinatlarının ön-nicemleme ile daha az duyarlılığa indirgenmesiyle çalışmaktadır. Bu temelle çalışan bazı yöntemler (DEERING, 1995) (TAUBIN&ROSSIGNAC, 1998) (TOUMA, 1998) (ISENBURG, 2002), kodlanan yüzeyin düzlük varsayımına dayanarak öngörü ile elde edilen düğüm koordinat öngörü hatalarını entropi kodlamaktadır. Sayıl nicemlemenin kullanıldığı bu öngörücü kodlama yöntemlerinden başka vektör nicemlemenin kullanıldığı öngörücü kodlama yöntemleri de (BAYAZIT, 2007) (LEE, 2000) (CHOU, 2002) önerilmiştir. Öngörüden yararlanan bütün bu yöntemlerde düğümlerden geçiş sırasını belirleyen bağlanırlık kodlaması geometri kodlaması için yardımcı araç vazifesi görmektedir. 3 boyutlu uzayda aşamalı iletim için tasarlanmış bazı geometri kodlayıcılar (HOPPE, 1996) (PAJAROLA, 2000) (TAUBIN&GUEZIEC, 1998) çoklu çözünürlük sağlamakla beraber geometri sıkıştırma için yeterli başarımları sağlayamamıştır.

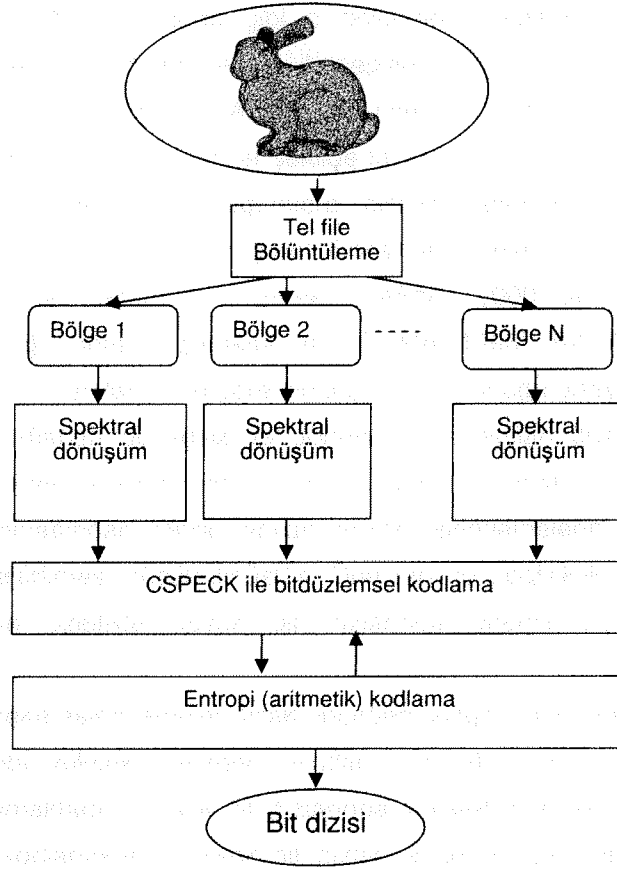
Ayrık dalgacık dönüşümünün (SZYMCZAK, 2002)'de önerilen yarı-düzenli yeniden örme (topolojik bilgiyi değiştirme) yaklaşımı ve (SAID, 1996)'daki küme bölüntüleme kodlama tekniği ile bütünleştirilmesi ile oluşturulan çok çözünürlüklü geometri sıkıştırma yöntemleri (KHODAKOVSKY, 2000) (KHODAKOVSKY,

2002) yüksek başarımları sağlayarak denektaşları yöntemleri olmuştur. Özellikle dalgacık tabanlı görüntü kodlamada kullanılmış olan bazı diğer algoritma ve kodlama tekniklerinin (KHODAKOVSKY, 2002)'de önerilen dalgacık dönüşümü ile bütünleştirilmesi ile daha verimli sıkıştırma yöntemleri (LAVU, 2003) (PAYAN, 2005) de elde edilmiştir.

Tel file geometri sıkıştırma için önerilen diğer bir dönüşüm (KARNI, 2000) ise düzensiz tel file topolojisinden türetilen birimlik taban vektörleri ile sağlanmaktadır. (KARNI, 2000)'de tel file koordinat vektörlerinin bu taban vektörleri üzerine izdüşürülmesiyle hesaplanan katsayıların sadece yüksek enerjili bir altkümü tutulmaktadır. Tutulan katsayılar sayıl nicemlenerek entropi kodlanmaktadır. Bu yöntem, tutulan katsayı altkümünün zaman içinde niceliğinin artırılmasıyla aşamalı iletme de olanak tanıdığı için bir ölçüde gömülü bit dizisi üretebilmektedir.

Bu projede önerilen kodlama sisteminde yüzeysel olarak bölüntülenmiş bir tel filenin her bölgesinde (KARNI, 2000)'deki dönüşüm yardımıyla hesaplanan spektral dönüşüm katsayıları, küme bölüntüleme yaklaşımıyla çalışan CSPECK (Color-Set Partitioning Embedded Block Coder (PEARLMAN, 2004)) algoritması ile genliklerinin büyüklük sıralarına göre işlenip kodlanmaktadır. CSPECK her tel file bölgesindeki her koordinat için gittikçe azalan eşik değerlerinde *önemli* katsayıları saptamakta ve bu katsayıların konum bilgisini ardışık küme önemlilik kararları şeklinde kodlamaktadır. Belli bir eşik değeri için önemli olarak saptanan katsayıların en önemli bitlerini gösteren ilk geriçatımları kodlamanın ileriki aşamalarında daha önemsiz bitlerinin de gösterilmesi ile inceltilmektedir.

Geliştirilen yöntem, koordinat ve tel file bölgelerine bit atamasını dolaylı olarak gerçekleştirildiğinden (KARNI, 2000)'deki yöntemin sağladığından daha doğru bir bit önceliği sağlamakta ve dolayısıyla daha gömülü bir bit dizisi üretebilmektedir. Bununla beraber, birden fazla önemsiz katsayı tek bir sembolle kodlanarak önemsiz katsayılar arasındaki genlik bağımlılıkları değerlendirildiği için de (KARNI, 2000)'deki yöntemle göre daha yüksek bir sıkıştırma başarımları sağlanmaktadır. Önerilen geometri sıkıştırma yönteminin şematik anlatımı Şekil 1'de sunulmaktadır.



Şekil 1 Önerilen dönüşüm tabanlı geometri sıkıştırma yönteminin şematik gösterimi

Şekil 1’de sunulan yöntemin geliştirilmesi için sembollerinin koşullu ve ortak entropi kodlanması araştırılmıştır. Sıralanan katsayılar ve dolayısıyla komşu katsayı kümelerinin önemlilik ve inceltme sembolleri arasındaki istatistiksel bağımlılıkların kuvvetli olması durumunda koşullu ve ortak entropi kodlamanın ek bir başarımlı artımı vermesi beklenebilir. Koşullu kodlama için geniş bir model kümesi üzerinden deneme yanılma ile uygun bağlamların varlığı da araştırılmıştır.

Yukarıda önerilen yöntemin (KHODAKOVSKY, 2002)’de önerilen yöntemle de kuvvetli bir alternatif oluşturduğu benzetimler yardımıyla gösterilmiştir. Ne var ki gerek (KHODAKOVSKY, 2002) (LAVU, 2003) (PAYAN, 2005) gibi yöntemlerin ilklendirme adımı olan yarı-düzenli yeniden örme işlemi gerekse de (KARNI, 2000)’deki dönüşüm için gerekli yüksek boyutlu matrisin özvektörlerini saptama işlemi yüksek hesaplama karmaşıklığına sahip olduğundan bu yöntemlerin pratik uygulanabilirlikleri sınırlıdır. Projede yapılan çalışmanın ikinci aşamasında dönüşüm esnasında karşılaşılan hesaplama karmaşıklığını düşük tutmak için (KARNI, 2001)’de önerilen değişmez spektral taban vektörleri ile gerçekleştirilen dönüşüm incelenmiştir. Kodlanan (düzensiz) tel file bölgesine uygulanan

dönüşüm için seçilmiştir. KARNI, 2000) tarafından geliştirilen yöntem

dönüşüm (KARNI, 2000)'de olduğu gibi o bölgedeki bağırlık bilgisinden türetilen Laplas işlecinin özvektörlerinin hesaplanmasıyla suretiyle bölgeye uyarlamalı biçimde oluşturulmayıp, (KARNI, 2001)'de önerildiği gibi, o bölgedeki düğümlerin, özvektörleri önceden benzer şekilde tasarlanmış uygun büyüklükteki bir düzenli tel file bölgesindeki düğümlere birebir eşlemlenmesi suretiyle oluşturulmuştur. Söz konusu yöntemin geliştirilmesi ve incelenmesi neticesinde bu yöntemin kodlama başarımının gerek (KARNI, 2000)'deki uyarlamalı dönüşümün kullanıldığı yöntem gerekse de (TOUMA, 1998) (BAYAZIT, 2007) gibi öngörüye dayalı yöntemlerinkine göre oldukça düşük olduğu ortaya çıkmıştır.

Çözüm olarak kodlanan düzensiz ve dönüşümü kullanılan düzenli tel file bölgelerindeki düğümler arasında dönüşümün uygulanmasından önce ve ters dönüşümün uygulanmasından sonra yapılan birebir eşlemlenmelerin çoktan bire eşlemlenmelere değiştirilmesi düşünülmüştür. Önerilen bu değişiklik neticesinde kodlama kazancının özellikle düşük bit hızlarında önemli ölçüde arttığı gözlemlenmiştir.

Önerilen çoktan teke eşlemlenmeler alçak geçirgen süzgeç gibi işlev gördükleri için köşe kenar gibi yüksek geçirgen ayrıtları dönüşüm ve ters dönüşüm esnasında korunmamaktadır. Düşük bit hızlarında yüksek enerjiye sahip alçak frekans katsayıları kodlanırken yüksek bit hızlarına geçildikçe düşük enerjiye sahip alçak frekans katsayıları kodlanmakta, kaybedilen yüksek frekans bilgisi kodlanmamaktadır. Dolayısıyla kodlama başarımı yüksek bit hızlarına geçilirken eniyilikten uzaklaşmaya başlamaktadır. Kodlama başarımının yüksek bit hızlarında da yüksek tutulabilmesi için her bölge için dönüşüm tabanlı kodlamanın sadece belli bir bit hızına kadar uygulanması ve bu bit hızına gelindiğinde ters dönüşüm alınarak daha yüksek bit hızlarına düğüm koordinat değerlerinin geriçatım hatalarının uzamsal kodlanması ile ulaşılması önerilmektedir.

Literatürde genelde rastlandığı gibi bu çalışmada geliştirilen yöntemler üçgen yüzlerden oluşan üçgen tel fileler içindir. Bu yöntemlerin ufak değişikliklerle daha genel çokgen yüzlere sahip tel filelere de uygulanmaları mümkündür.

2. Bölümde proje çalışmasında kullanılan ve incelenen, literatürde önerilmiş olan tel file yüz bölüntüleme, dönüşüm ve kodlama yöntemleri tanıtılacaktır. 3. Bölümde (KARNI, 2000)'de önerilen uyarlamalı dönüşümün CSPECK tabanlı küme bölüntüleme yöntemi ile bütünleştirilmesi ile gerçekleştirilen tel file geometri sıkıştırma yöntemi anlatılıp kodlama başarımının eniyileştirilmesi için üretilen sembollerin koşullu ve ortak entropi kodlamasından bahsedilecektir. Bu bölümde ayrıca proje çalışmasının ikinci aşamasında geliştirilip önerilen,

dönüşüm için değişmez spektral taban vektörlerinin kullanıldığı yöntem de anlatılacaktır. Bu yöntemde her tel file bölgesi için dönüşüm tabanlı kodlamanın sonlandırılarak uzamsal kodlamaya geçilmesi için kullanılan karar verme mekanizmasına da değinilecektir. 4. Bölümde, önce proje çalışmasının ilk kısmında (KARNI, 2000)'de önerilen uyarlamalı dönüşüm kullanılan CSPECK kodlama yönteminin standart test modellerine uygulanması ile elde edilen sonuçlar sunulmaktadır. Karşılaştırma için test modellerinin (TOUMA, 1998), (KARNI, 2000) ve (KHODAKOVSKY, 2002)'de önerilen kodlama yöntemleri ile elde edilen sıkıştırma sonuçları da verilmektedir. Bu bölümde ikinci olarak da (KARNI, 2001)'de dönüşüm ve ters dönüşüm esnasında hesaplama karmaşıklığını azaltmak için önerilmiş olan değişmez taban vektörleri kullanılan yöntem ve bu yöntemin bu proje çalışması kapsamında çoktan bire eşlemler ile önerilen iyileştirilmiş şekli test modelleri üzerinde benzetimler yapılarak birbirleri ile ve (KARNI, 2000)'deki uyarlamalı dönüşüme dayanan yöntem ile karşılaştırılmaktadır. 4. Bölümde, son olarak, geliştirilen yazılım gerçeklemeleri üzerinden bu yöntemlerin yürütüm süreleri de bildirilmektedir.

Bekir Z. Özyürek, A. Özyürek, A. Özyürek
modellerinin sıkıştırma sonuçları

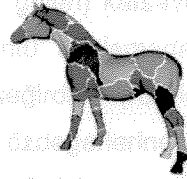
2.2 Spektral Analiz

2.2.1 Spektral Analiz

2. Genel Bilgiler

2.1 Tel file yüzey bölüntüleme

Projede uygulanan uyarlamalı ve değişmez taban vektörler ile sağlanan spektral dönüşümlerin orta-büyük (>10000 düğüm) bir tel filenin bütün düğümleri için hesaplanmaları maliyeti oldukça yüksek işlemlerdir. Z düğümlü bir tel file için hesaplama karmaşıklığının $O(Z^3)$ olduğu gösterilebilir. Bu duruma getirilebilecek pratik bir çözüm orta-büyük bir tel fileyi küçük tel fileler bölgelerine (<1000 düğüm) bölüntülemek ve herbir bölge için ayrı bir dönüşüm hesaplamaktır (KARNI, 2000). Literatürde çok seviyeli graf bölüntüleme için çok sayıda yöntem bulunmaktadır. Temel prensipte bu yöntemler herbir bölüntü parçasının bir işlemci ile işleneceği düşüncesinden yola çıkarak işlemcilere düşen yük dağılımını dengelemeye ve işlemciler arasındaki iletişim mesajlarını (kenar kesme) en aza indirmeye yönelik çalışırlar. (KARNI, 2000)'de olduğu gibi çalışmamızda MeTiS (KARYPIS, 1998) yazılım paketi tel file bölüntüleme için hızlı bir çözüm sağladığından kullanılmıştır. Tel file bölüntüleme, bölgelerin birleşim noktalarında DCT tabanlı görüntü kodlamada rastlanan bloklama etkisine benzer bir bozulmaya yol açmaktadır. Bununla beraber bölgeye uyarlamalı dönüşüm tasarımı tel filenin bölgesel özelliklerinin daha iyi gösterilerek verimli kodlanmasını sağlamaktadır. Şekil 2'de 19,851 düğümden oluşan *horse* modelinin MeTiS ile 40 bölgeye ayrıldıktan sonra bölgelere göre renklendirilen yüzey kaplaması gösterilmektedir.



Şekil 2 MeTiS sıradüzensel bölüntüleme yazılımı ile 40 bölgeye ayrılan *Horse* modelinin bölgelere göre renklendirilen yüzey kaplaması

2.2 Spektral dönüşüm

(KARNI, 2000)'de önerilen spektral dönüşüm, bir tel file bölgesindeki düğümlerin x, y veya z koordinatlarından birinin değerlerinden oluşan vektörün birimdik taban vektörleri üzerine izdüşürülerek o koordinat için spektral katsayı vektörünün

hesaplanmasına dayanmaktadır. Farklı koordinatlar ve bölgeler için dönüşümler birbirinden bağımsız olarak uygulanmaktadır. Bir bölgenin üç koordinat vektörünün üzerine izdüşürüldüğü birimlik taban vektörleri, o bölgedeki tel file topolojisinden türetilen Laplas işlecinin özvektörlerinden oluşmaktadır.

İndisi i olan düğümün derecesini (komşu düğümlerinin sayısını) d_i ile gösterirsek tel file bölgesinin Laplas işleci L 'nin elemanları

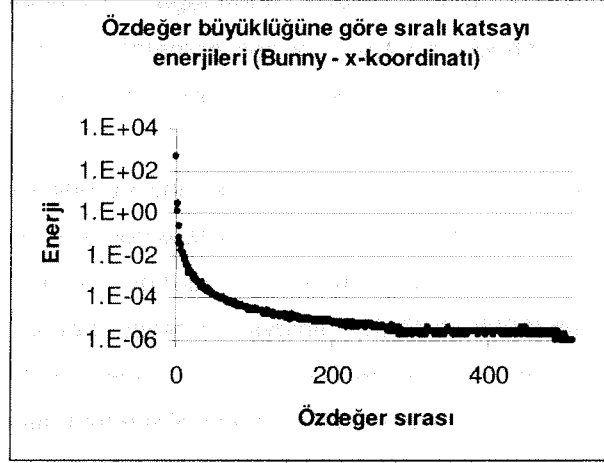
$$L_{ij} = \begin{cases} 1 & i = j \\ d_i^{-1} & i \text{ ve } j \text{ komşu ise} \\ 0 & \text{diğer} \end{cases} \quad (1)$$

olarak tanımlanmaktadır. 3 boyutlu tel file bölgesinin geometrisi için birimlik bir taban vektör kümesi teşkil eden L 'nin özvektörleri R^n 'i germektedir. Özvektörler $L\psi_i = \lambda_i\psi_i$ denkleminde bulunur ve taban vektörleri olarak kullanılır. Her koordinat $\gamma \in \{x, y, z\}$ için koordinat vektörü s_γ , tel file (bölgesindeki) düğüm noktalarının γ koordinatlarının gruplandırılması ile elde edilir. Katsayı vektörü c_γ , koordinat vektörünün taban vektörlerine ($\Psi = [\psi_1 | \dots | \psi_N]^T$) izdüşürülmesi ile $c_\gamma = \Psi s_\gamma$ olarak elde edilir.

Bu özvektörlere ait özdeğerler tel file geometrisine ait sinyalin frekansları olarak düşünülebilir. Düz tel file bölgelerinde, başka bir deyişle düğüm koordinatları komşu düğüm koordinatlarının ortalama değerine yakın olduğunda, küçük özdeğerli özvektörler kullanılarak hesaplanan düşük frekans katsayıları baskın olup enerjinin büyük bölümü bu katsayılarla taşınmaktadır. Düz olmayan tel file bölgelerinde ise öngörü hatasının gösterilebilmesi için çok daha fazla sayıda katsayıya ihtiyaç duyulmaktadır. Düz bölgelere sahip bir tel filenin az sayıda (küçük özdeğere karşılık gelen) katsayı ile gösterilerek yüksek kalitede kodlanması, DCT tabanlı görüntü kodlamada yüksek frekans katsayılarının sıfırlanıp verimli bir şekilde seğırdim (run-length) kodlanmasına benzerlik taşımaktadır. Bunny modeli için özdeğerlerine göre küçükten büyüğe sıralanan katsayıların enerjilerinin azalışı Şekil 3'te gösterilmektedir. Gerek (KARNI, 2000)'deki yöntemin uygulanması gerekse de projede önerilen yöntemin uygulanması esnasında katsayıların büyüklükleri arasındaki ilintiyi enbüyüterek eniyi sıkıştırma kazancı sağlamak için katsayılar özdeğerlerine göre küçükten büyüğe sıralanmışlardır.

Her ne kadar teoride spektral dönüşüm matrisi Ψ dikgen ($\Psi\Psi^T = I$) olsa da yüksek boyutlu L matrisinden sonlu kesinliğe sahip işlemlerle hesaplanan Ψ pratikte dikgen olmaktan uzaktır. Bu yüzden dönüşüm tabanlı sıkıştırmadaki tam

geriçatım özelliğini sağlamak için ters dönüşüm aşamasında Ψ^T yerine Ψ^{-1} kullanılması uygun görülmüştür.



Şekil 3 Özdeğerler küçükten büyüğe doğru sıralandığında karşılık gelen katsayıların enerjilerinin hızla azaldığı gözlemlenmektedir.

Kodçözücüde uygulanan Ψ^{-1} 'nin hesaplanabilmesi için gerekli L matrisi bağlanırlık bilgisinin kodlayıcıdan kodçözücüye kayıpsız bir şekilde gönderilmesi ile bulunabilir. Bu çalışmada bağlanırlık bilgisinin kodçözücüde bulunduğunu varsayıyoruz.

2.3 Spektral tel file sıkıştırma yöntemi

(KARNI, 2000)'deki spektral sıkıştırma yönteminde spektral dönüşümün uygulanması ile her koordinat γ için elde edilen katsayılar vektörü c_γ 'nin (kırılmak suretiyle elde edilen) en küçük özdeğerlere karşılık gelen katsayılarını taşıyan altvektörü, vektör bileşenlerinin sayı nicemlenmesi ve niceme seviyelerinin entropi kodlanması ile kodlanmaktadır. Kodlanan her katsayı n bit ile nicemlenirken bir bit işaret, $n-1$ bit ise genlik gösterimi için kullanılır. Altvektör oluşturmadaki rasyonel yukarıda bahsedildiği gibi küçük özdeğerlere karşılık gelen katsayıların enerjilerinin genelde büyük olmasıdır.

Her ne kadar eski JPEG standardı gibi dönüşüm tabanlı bazı kodlama sistemlerinde de yüksek enerji beklentili katsayıların bitlerine bit dizisi içinde öncelik verilerek kodlama yapılırsa da, iki nedenden ötürü (KARNI, 2000)'deki yöntem yeterli sıkıştırma kazancı sağlayamamaktadır. Bunlardan birincisi katsayılara doğru bit atamasının yapılmamasıdır. Yüksek enerjisi olması beklenen katsayıların gerçekten yüksek enerjili olup olmadıklarına bakılmadan,

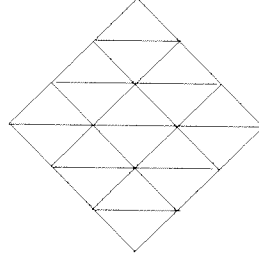
en önemsiz dahil bütün bitlerinin, düşük enerjisi olması beklenen katsayıların bütün bitlerine göre öncelikli olarak bit dizisine yerleştirilmesi doğru bir yaklaşım değildir. İkinci neden de arasında değerlendirebilecek ilintiler olan katsayıların ortak veya birbirlerinin değerlerine koşullu olarak kodlanmamasıdır. Proje çalışmasında önerilen yöntemde her iki husus da dikkate aldığı için spektral tel file sıkıştırma yöntemine göre daha yüksek sıkıştırma başarımı sağlanacağı öngörülmüştür.

(KARNI, 2000)'deki sıkıştırma yönteminin uygulanmasında karşılaşılan pratik bir problem sayıl nicemleme seviyelerinin çokluğundan dolayı büyük bir kaynak alfabeti kullanılarak entropi kodlama yapmanın güçlüğü ve alfabe büyüklüğünden doğan başarımların kayıptır. (KARNI, 2000)'de kullanılan entropi kodlama yönteminin detaylı açıklaması verilmediğinden (WITTEN, 1987)'de önerilen uyarlamalı aritmetik kodlama kullanılmıştır. Büyük alfabeler için aritmetik kodlamada kullanılan olasılık modellerinin başlangıç durumundan son duruma yakınsamaları çok sayıda sembol kodlaması gerektirdiğinden 14-16 bit seviyeli nicemleyicinin çıktısını uyarlamalı aritmetik kodlama sadece yüzbinlerce düğümüne sahip modeller için makul bir başarımlar sağlamaktadır.

2.4 Değişmez taban vektörleri kullanılarak düzensiz tel file geometrilerinin kodlanması

Bölüm 2.2 ve 2.3'de bahsedilen tel file geometri kodlama yönteminde her tel file bölgesinin kodlanması için kullanılan dönüşüm taban vektörleri o tel file bölgesinin bağlantı bilgisi kullanılarak elde edilen Laplas işlecinin özvektörleri olarak alınmaktadır. Dönüşüm taban vektörlerinin böyle her model bölgesine uyarlamalı olarak belirlenmesi, yüksek sıkıştırma kazancı sağlamakla beraber özellikle kodlayıcıda yüksek hesaplama karmaşıklığına neden olmakta, ve yöntemin hızlı kodlama gerektiren bir uygulamada yer bulmasını engellemektedir.

(KARNI, 2001)'de çözüm olarak dönüşüm taban vektörleri olarak altıgen düzenli bir tel filenin Laplas işlecinin özvektörlerinin kullanılması önerilmiştir. Şekil 4'de $N \times N$ 'lik ($N=4$) altıgen düzenli bir tel file bölgesi örneği gösterilmektedir.



Şekil 4 4x4'lük düzenli altıgen tel file bölgesi

(KARNI, 2001)'de önerilen yöntemde kodlama öncesinde çeşitli büyüklükte ($N=5, \dots, 30$) düzenli altıgen tel file bölgeleri için taban vektör kümeleri tasarlanmaktadır. Kodlanacak düzensiz tel file bölgesinin iç düğüm sayısı ve sınır düğüm sayısı bulunmakta, iç ve sınır düğüm sayıları en az bu sayılar olan en küçük düzenli altıgen tel file bölgesi saptanmakta ve bu tel file bölgesi için tasarlanmış olan taban vektörleri dönüşümde kullanılmaktadır.

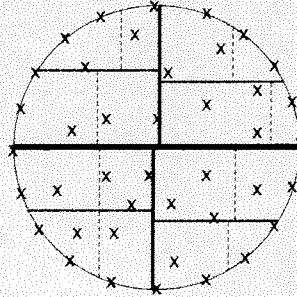
Dönüşümün gerçekleştirilebilmesi için kodlanan düzensiz tel file bölgesindeki düğümlerin saptanmış olan düzenli tel file bölgesindeki düğümlere eşlemlenmesi gerekmektedir.

Bire bir eşleme yapılan bölgelerdeki iç ve sınır düğüm sayılarının eşit olması için kodlanan düzensiz tel file bölgesine düğümler ilave edilir. Bu noktada ortalama değerliliği en az olan yüzler arasından değerlilik varyansı en düşük yüze düğümler ilave edilmesi önerilmiştir. Çalışmamızda ise en büyük değerliliği en küçük olan yüze düğümlerin ilave edilmesinin başarıyı etkilemeden yazılım sisteminin hızını artırdığı gözlemlenmiştir. İlave edilen düğümlerin komşulukları (tel file bölgesi içinde ve dışında) ve ilave edilen düğümlere komşu (tel file bölgesi içinde ve dışında) olan düğümlerin komşulukları güncellenmektedir. Ayrıca tel file bölgesinde oluşan yeni yüzler (üçgenler) de kayıt edilmektedir.

Eşleme için (KARNI, 2001)'de önerilen geometrik yöntemde önce düzenli altıgen tel file bölgesinin sınır düğümleri birim çember üzerinde eşit açılarla yer alan noktalar ile temsil edilir. Düzenli altıgen tel file bölgesinin iç düğümleri birim dairede rastgele noktalar olarak iklendirilir ve özyineli olarak her nokta komşularının aritmetik ortalaması olacak şekilde birim dairedeki nihai nokta konumları saptanır.

Benzer şekilde kodlanan düzensiz tel file bölgesindeki sınır düğümler de birim çember üzerinde eşit açılarla yer alan noktalar ile temsil edilir. Düzensiz tel file bölgesinin iç düğümleri birim dairede rastgele noktalar olarak iklendirilir ve özyineli olarak her nokta komşularının aritmetik ortalaması olacak şekilde birim dairedeki nihai nokta konumları saptanır.

İki birim daire üzerindeki çok sayıda (~500) noktanın birbirine geometrik olarak eşleşmesi oldukça hesaplama maliyeti yüksek bir işlem olduğundan (KARNI, 2001)'de bahsedilen böl ve yönet yaklaşımı uygulanmıştır. Buna göre her iki birim dairedeki noktalar özyineli olarak bölüntülenir. Her özyineleme adımında her iki birim dairedeki noktalar değişimli olarak düşey veya yatay çizgilerle iki altkümeye bölüntülenmektedir. Sağda kalan altküme soldaki altküme kadar veya bir fazla nokta içermekte, benzer şekilde, alttaki altküme üstteki altküme kadar veya bir fazla nokta içermektedir. Şekil 5'te bu bölüntüleme işlemi örnek bir nokta kümesi üzerinde gösterilmektedir. İki birim dairelerde birbirine karşılık gelen en küçük altkümelere içindeki noktalar (ve dolayısıyla bu noktaların temsil ettiği tel file bölgelerindeki düğümler) *birebir* eşleşmektedir. Burada $M(j)$ düzenli altıgen tel file bölgesindeki j düğümünün kodlanan düzensiz tel file bölgesinde eşleştiği düğümü gösterebilir. (KARNI, 2001)'deki gibi iki altkümedeki noktaların eşleşmesinde toplam karesel uzaklığın en aza indirgenmesi amaçlanmakla beraber (KARNI, 2001)'de kullanılan fırsatçı atama algoritmasının yerine atama probleminin en iyi çözümü olarak bilinen Macar yönteminin (Hungarian method) uygulanması tercih edilmiştir.



Şekil 5 Birim dairedeki örnek nokta kümesinin altkümelere bölüntülenmesi. Her özyineleme adımında noktalar yaklaşık eşit büyüklükteki iki kümeye bölüntülenmektedir. Her alt bölüntülemeyi gösteren çizgilerin kalınlıkları azalmaktadır. En alt seviyedeki bölüntüleme kesikli çizgilerle gösterilmektedir. En alt seviyedeki her altküme iki veya üç nokta içermektedir.

Dönüşüm ve kodlama öncesinde kodlanan tel file bölgesindeki düğümlerin y koordinatından oluşturulan koordinat vektörü s_y 'nin bileşenleri doğru permütasyon ile $\hat{s}_{y,j} = s_{y,M(j)}$ vektörüne dönüştürülmektedir. \hat{s}_y vektörünün saptanan uygun büyüklükteki altıgen düzenli tel file bölgesinin Laplas işlecinin

özvektörlerine izdüşümü neticesinde γ koordinatı için bir katsayı vektörü c_γ elde edilmektedir. Bütün tel file bölgelerinin bütün koordinatları için katsayı vektörleri önceki bölümde anlatılan tel file sıkıştırma algoritması ile kodlanmaktadır.

CSPECK kodçözme sonucunda elde edilen \hat{c}_γ geriçatılmış katsayıları içermektedir. Dönüşümde kullanılan matrisin tersinin uygulanması ile elde edilen vektör \hat{s}_γ , doğru permütasyon ile $\hat{s}_{\gamma,j} = \hat{s}_{\gamma, M^{-1}(j)}$ vektörüne dönüştürülüp kodlanan tel file bölgesindeki düğümlerin γ koordinatlarının geriçatımları elde edilmektedir.

Geriçatılan tel filede ilave edilen düğümler de bulunacağından bozunum hesaplanırken bu düğümler hesaba katılmamaktadır. Ne var ki düğümlerin ilave edilmesi katsayı vektörünün boyutunu da artırdığından bit hızının artması kaçınılmazdır.

3. Gereç ve yöntem

3.1 CSPECK algoritmasının tel file geometri sıkıştırılmaya uygulanması

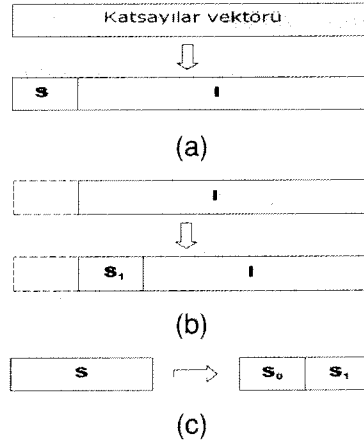
CSPECK'in ana gövdesi SPECK (ISLAM, 1999), bir denektaşı dalgacık tabanlı görüntü kodlama yöntemi olarak kabul edilen SPIHT (SAID, 1996) algoritmasının basit bir biçimidir. Dalgacık tabanlı görüntü kodlamada altbantlar arası katsayı genlik ilintilerini değerlendirmeyi de hedefleyen SPIHT'in aksine SPECK sadece altbant içi katsayı genlik ilintilerini değerlendirmeyi hedefler. Belli bir genlik eşğine göre önemlilik arz eden katsayıların konumları altbant içi katsayı kümelerinin özyineli olarak bölüntülenmesi ile saptanır ve kod çözücü tarafa belirtilir. Önemsiz katsayıların sıfır değerli önemli bitleri arasındaki ilintiler, bu bitler kümeler halinde gruplandırılarak, ve ortak olarak tek bir sembole kodlanarak değerlendirilir.

CSPECK ile tel file bölge dönüşüm katsayıları kodlanmaya başlanmadan önce her bir γ koordinatı için elde edilen katsayılar küçük özdeğerlerden büyük özdeğerliye doğru sıralanır ve c_γ katsayılar vektörü elde edilir. Katsayılar vektörünün i 'nci katsayısı $c_{\gamma,i}$ olarak gösterilsin. Ayrıca bir katsayı kümesi T 'nin n değerine göre önemlilik gösterge işlevi

$$\Pi_n(T) = \begin{cases} 1 & 2^n \leq \max_{i \in T} \{|c_{i,\gamma}|\} < 2^{n+1} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (2)$$

olarak tanımlansın. Başka bir deyişle n değerine göre önemlilik arz eden T kümesindeki en az bir katsayının ikili gösterimde ondalık noktasından önceki $(n+1)$ 'inci biti bir, daha önceki bitleri sıfırdır.

SPECK, bir katsayılar vektörünü oluşturan katsayılar kümesini sıradüzensel bir yapıda bölüntüler. Şekil 6a ilk bölüntüleme düzeyinde katsayılar vektörünün S ve I tipi kümeler olarak adlandırılan (alt)kümelere bölüntülenmesini göstermektedir. I tipi kümeler sonraki geçişlerde benzer biçimde Şekil 6b'de gösterildiği gibi ikiye bölüntülenmektedir. S tipi kümeler parçalandığında ise eşit boyda iki yeni S tipi küme oluşmaktadır (Şekil 6c).



Şekil 6 S ve I Tipi Kümelerin Bölüntülenmesi

SPECK'in algoritmik yapısında henüz herhangi bir n değerine göre önemli bulunmamış S tipi kümeleri tutan önemsiz kümeler listesi (ÖKML) ve önem testini geçebilmiş katsayıları tutan önemli katsayılar listesi (ÖKSL) adıyla iki tip liste bulunmaktadır.

SPECK ve CSPECK her seferinde n değerinin bir azaltıldığı kodlama geçişlerinden oluşur. Tek bir katsayılar vektörü için çalışan SPECK, CSPECK'e x , y ve z koordinatlarına ait üç tane katsayılar vektörünü bir arada kodlamak üzere genelleştirildiğinde, her kodlama geçişinde üç katsayılar vektörüne ait kümeler birbirlerinin işlenişinden bağımsız olarak işlenir.

CSPECK başlarken $n = n_{\max} = \left\lceil \log_2(\max_{\gamma,i} \{ |c_{\gamma,i}| \}) \right\rceil$ olarak ilklendirilir ve çıktılanır. Ayrıca eldeki ilk katsayılar vektörleri S ve I tipi ikişer kümeye bölüntülenir (Şekil 6a), üç S kümesi ÖKML'ye eklenir ve ÖKSL de boş olarak ilklendirilir.

Her kodlama geçişi üç adımdan oluşmaktadır. *Sıralama* adımında, ÖKML'lerdeki S tipi kümeler artan boy sırasına göre özyineli olarak işlenir. Ardından I kümesi boş değilse o da özyineli olarak işlenir. S 'nin işlenmesi önemlilik testindeki kararın çıktılanıp S 'nin önemli olması durumunda iki yeni S tipi kümeye bölüntülenmesi olarak gerçekleşir. S tipi kümelerin işlenmesi önemli katsayılara veya önemsiz S tipi kümelere varılana kadar devam eder. I 'nin işlenmesi ise önemlilik testinin sonucunun çıktılanıp I 'nin önemli olması durumunda bir yeni S tipi küme ve daha küçük bir yeni I tipi kümeye bölüntülenmesi olarak gerçekleşir. Önemsiz kümeler ÖKML'ye eklenir. Sıralama adımında ayrıca önemli katsayıların işaret sembolleri kodlanır. Bu katsayılar ÖKSL'ye ilk gerçeğatımları $\hat{c}_{\gamma,i} = \pm(1.5)2^n$ ve hataları $e_{\gamma,i} = c_{\gamma,i} - \hat{c}_{\gamma,i}$ kayıt edilerek

eklenir. İyileştirme adımında, son sıralama adımından önce önemli olduğu saptanan ÖKSL'deki katsayıların yeni biti $b_{\gamma,i} = (e_{\gamma,i} > 0)$ çıktılanır. Geriçatımları $\hat{c}_{\gamma,i} = \hat{c}_{\gamma,i} + (2b_{\gamma,i} - 1)2^{n-1}$ ve hataları $e_{\gamma,i} = c_{\gamma,i} - \hat{c}_{\gamma,i}$ olarak güncellenir. Nicemleme adımında ise bir sonraki geçiş öncesi n güncellenir ($n \leftarrow n-1$). ÖKML ve ÖKSL'nin elemanlarının bu sıra ile işlenişi istenilen bit hızı ya da bozunuma ulaşıncaya kadar sürdürülür. CSPECK küme bölüntüleme tabanlı kodlama algoritmasının sözde programı, altprogramları ile beraber aşağıda sunulmaktadır.

Algoritma:

1) İlkendirme

- Her koordinat için girdilene tel file (veya tel file bölgesi) üzerinde spectral dönüşümü uygula ve spektral katsayıları hesapla.
- Her koordinat için katsayıları özdeğerlerinin artış sırasına göre sırala ve katsayılar vektörünü elde et.
- Her koordinat için katsayılar vektörü c_{γ} 'yi S tipi ve I tipi olmak üzere iki kümeye bölüntüle.
- $n_{\max} = \left\lfloor \log_2(\max_{\gamma,i} |c_{\gamma,i}|) \right\rfloor$ değerini çıktıla. $n = n_{\max}$.
- Her koordinat için S tipi kümeyi ÖKML 'ye ekle ve ÖKSL = \emptyset olarak ilkendir.

2) Sıralama adımı

- Artan büyüklük sırasına göre ÖKML'deki her S tipi küme için $ProcessS()$.
- Eğer I tipi küme henüz boş değilse, $ProcessI(I)$.

3) İnceltme adımı

- Son sıralama adımında eklenenler hariç ÖKSL 'deki her katsayı $c_{\gamma,i}$ için $|c_{\gamma,i}|$ 'in n 'inci en önemli bitini çıktıla.

4) Nicemleme adımı

- n 'yi bir azalt ve adım 2'ye dön.

Altprogram $ProcessS(S)$

- $\prod_n(S)$ çıktıla.
- Eğer $\prod_n(S) = 1$
 - Eğer S tek katsayılık bir kümeysse, S 'nin işaretini çıktıla ve S 'yi ÖKSL'ye ekle.
yoksa $CodeS(S)$.
 - Eğer $S \in \text{ÖKML}$ ise S 'yi ÖKML'den çıkart.
yoksa
 - Eğer $S \notin \text{ÖKML}$ S 'yi ÖKML'ye ekle.
- Geri dön.

Altprogram $CodeS(S)$

- S 'yi eşit büyüklükte iki küme olan S_1 ve S_2 'ye bölüntüle.
- Her küme S_i ($i=1, 2$) için
 - $\prod_n(S_i)$ 'yi çıktıla.
 - Eğer $\prod_n(S_i) = 1$

- i. Eğer S_i tek katsayı kümesi ise, işaretini çıktıla ve ÖKSL'ye ekle.
yoksa $CodeS(S_i)$.
yoksa S_i 'yi ÖKML'ye ekle.
- 3) Geri dön.

Altprogram ProcessI(I)

- 1) $\prod_n(I)$ 'yi çıktıla.
- 2) Eğer $\prod_n(I) = 1$ ise
 - a. $Codel(I)$.
- 3) Geri dön.

Altprogram Codel(I)

- 1) I 'yi S tipi S_i ve I tipi ve I_i kümelerine bölüntüle.
- 2) $ProcessS(S_i)$.
- 3) $ProcessI(I_i)$.
- 4) Geri dön.

CSPECK kodlayıcı her bit düzlemi n için tüm bölgelerin tüm koordinatlarına ait katsayıları bir geçişte beraber gözden geçirdiğinden bölge ve koordinatlara dolaylı bit ataması yapıldığı söylenebilir.

3.2 CSPECK sembollerinin aritmetik kodlanması.

CSPECK algoritması tarafından önemli katsayıların işaret sembolleri işaret dağılımının bir biçim olmasından dolayı ikili kodlanmaktadır. Önemlilik testi sembolleri de ikili kodlanabileceği gibi önemsizlik durumunun yüksek olasılığa sahip olması özelliği değerlendirilerek sıkıştırma kazancı elde edilebilmesi için entropi kodlanabilirler. Benzer şekilde (BAYAZIT, 2003)'de olduğu gibi iyileştirme aşamasında çıktılanan bitler entropi kodlanarak küçük ama azımsanmayacak bir kazanç elde edilebilir. Buna göre büyük genliğe doğru yapılan iyileştirme bir sembolle, küçük genliğe doğru yapılan iyileştirme başka bir sembolle gösterilip entropi kodlanır.

Sembol alfabeleri küçük, ve dağılımları da kayık olduğundan Huffman kodlama yerine aritmetik kodlama tercih edilmiştir. Sembol harf olasılıkları önceden bilinmediğinden ve olasılıkları kestirmek için kodlayıcı tarafından bir ön geçiş yapma gecikmeye sebep olacağından model verisi üzerinden tek bir kodlama geçişi esnasında olasılıkların gözlemlenmiş harf sıklıklarından uyarlanır olarak (WITTEN, 1987) ve (SAID, 2004)'deki gibi kestirilmesi uygun bulunmuştur. Aritmetik kodlayıcı yazılımı olarak (SAID, 2004)'de geliştirilen algoritmaların hazır C kaynak kodları kullanılmıştır.

3.3 Geliştirilen ikili ve aritmetik kodlayıcılar için kod çözücülerin gerçekleştirilmesi.

Son iki bölümde anlatılan kodlayıcılar için kodçözücüler de gerçekleştirilmiştir. Bir model için kodlama sonucu elde edilen bit dizisi kod çözücüye girilerek kod çözücünün çıktığı gerçeğin kalitesine bakılmış, bu kalitenin bit hızı ile artan nitelikte ve kodlayıcı taraftakine eşdeğer olduğu gözlemlenerek kodçözücünün doğrulaması yapılmıştır.

Aritmetik kodlama kullanıldığı takdirde kod çözücüde bit dizisi ilk olarak aritmetik kodçözücüye girilir. Aritmetik kod çözücü önemlilik, işaret ve iyileştirme sembollerini oluşturur. Önemlilik sembollerini girdiye küme bölüntüleme kod çözücüsü, kodlayıcı ile aynı durum makinasını gerçekleştirerek aynı sırada katsayı kümelerini bölüntüler. Sıralama adımlarında önemli bir katsayı için kodçözülen işaret sembolü ile ilk gerçeğin $\hat{c}_{\gamma,i} = \pm(1.5)2^n$ olarak yapılırken iyileştirme adımlarında gerçeğin $\hat{c}_{\gamma,i} = \hat{c}_{\gamma,i} + (2b_{\gamma,i} - 1)2^{n-1}$ olarak güncellenir.

Bölüm 2.2'de bahsedildiği gibi geri çarılan katsayılara ters dönüşüm olarak Ψ^T yerine Ψ^{-1} uygulanması uygun bulunmuştur.

3.4 Basit bir örnek

Bu bölümde 4 düğüm ve 4 üçgen yüze sahip basit bir tel file üzerinde kodlama yöntemini sembollerin entropi kodlamasına fazla değinmeden anlatmak istiyoruz. Burada amacımız izlenen işlemlerin sırasını vurgulamak ve kodlanan sembellere değinmektir.

Şekil 7'de VRML dilinde basit bir tel file örneği gösterilmektedir. Düğüm koordinatlarının listelenmesi ile 4×3 'lük geometri matrisi G 'nin oluşturduğu düşünülebilir.


```

#VRML V2.0 utf8
Shape {
  geometry IndexedFaceSet {
    coord Coordinate {
      point [
        -3.0  2.0  2.0
         7.0 -1.0  9.0
         2.0  8.0  7.0
         8.0  3.0  5.0
      ]
    }
    coordIndex [
      0 1 2 -1
      2 3 0 -1
      2 1 3 -1
      0 3 1 -1
    ]
  }
}

```

Şekil 7 VRML dilinde bir tel file örneği

Tel filenin komşuluk matrisi N (ij elemanı i ve j indisli düğümlerin komşu olup olmadığını gösterir) ve Laplas işleci L

$$N = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad \text{ve} \quad L = \begin{bmatrix} 1.000 & -0.333 & -0.333 & -0.333 \\ -0.333 & 1.000 & -0.333 & -0.333 \\ -0.333 & -0.333 & 1.000 & -0.333 \\ -0.333 & -0.333 & -0.333 & 1.000 \end{bmatrix}$$

olarak yazılabilir.

x , y and z koordinatları için spektral dönüşüm katsayılarını hesaplamak için, Laplas işlecinin sütun özvektörleri hesaplanır. Bu özvektörler ve karşılık gelen özdeğerler

$$\Psi = \begin{bmatrix} -0.500 & -0.216 & 0.866 & 0.089 \\ -0.500 & 0.839 & -0.289 & -0.345 \\ -0.500 & -0.146 & -0.289 & -0.520 \\ -0.500 & -0.478 & -0.289 & 0.776 \end{bmatrix}$$

matrisi ve

$$\Lambda = \begin{bmatrix} 0.000 & 1.333 & 1.333 & 1.333 \end{bmatrix}$$

satır vektörü olarak hesaplanır. Ψ matrisinde özvektörler özdeğerlerinin artış sırasında sıralanmıştır.

Tablo 1'de sunulan üç koordinat bileşeni için spectral katsayılar (satır vektörleri $G^T \Psi$ olarak hesaplanmış olup G 'nin sütunlarının özvektörlere izdüşümleridir.

x	-7.000	2.408	-7.506	2.491
y	-6.000	-3.868	-1.155	-1.311
z	-11.500	3.712	-4.330	-2.685

Tablo 1 x, y ve z için katsayılar satır vektörleri

Kodlamayı anlatmaya başlamadan önce iklendirme safhasına ait bazı gözlemlere değinelim. c_x , c_y ve c_z üç koordinatın 1x4'lük spectral katsayılar vektörleridir. En büyük katsayı büyüklük değeri 11.500'dir. İki tabanında en büyük büyüklüğe sahip katsayıyı gösteren en büyük n değeri $n_{max} = \lfloor \log_2 11.500 \rfloor = 3$ dir. Önerilen kodlayıcı katsayı büyüklükleri için düzelenmiş bir tanım kümesinde çalışmaktadır. Örneğin, en büyük büyüklüklü katsayının n_{max} 'e göre bu tanım kümesinde değeri 1.4375'dir. Bu katsayının gerçek büyüklüğü $2^{n_{max}} \times 1.4375 = 11.500$ olarak elde edilir.

Tablo 2'de, kodlamanın beş ardışık geçişte nasıl gerçekleştiğini göstermekteyiz. Her geçiş için gösterilen değerler o geçişin sonunda elde edilen değerlerdir. Tabloda, her kodlama geçişinde, işlenen bit düzlemini gösteren n değeri, katsayı büyüklüklerinin önemini sınamada kullanılan eşik T_n , katsayıların önemlilik durumları, o geçişte önemli bulunan katsayıların kodlanan işaret sembolleri ve o geçiş sonrasında katsayıların geriçatılan büyüklükleri verilmektedir.

Geçiş 1 Bit düzlemi $n = 3$ Eşik $T_1 = 5.750$	Önemlilik durumu				
	x	1	0	1	0
	y	1	0	0	0
	z	1	0	0	0
Önemli bulunan İşaret sembolü					
x	-			-	
y	-				
z	-				
Geriçatılan büyüklükler					
x	8.625	0.000	8.625	0.000	
y	8.625	0.000	0.000	0.000	
z	8.625	0.000	0.000	0.000	
Geçiş 2 Bit düzlemi $n = 2$ Eşik $T_2 = 2.875$	Önemlilik durumu				
	x	1	0	1	0
	y	1	1	0	0
	z	1	1	1	0
Önemli bulunan İşaret sembolü					
x					
y			-		
z		+		-	
Geriçatılan büyüklükler					
x	7.188	0.000	7.188	0.000	
y	7.188	4.313	0.000	0.000	
z	10.063	4.313	4.313	0.000	

<p>Geçiş 3</p> <p>Bit düzlemi $n = 1$</p> <p>Eşik $T_3 = 1.438$</p>	Önemlilik durumu				
	x	1	1	1	1
	y	1	1	0	0
	z	1	1	1	1
Önemli bulunan İşaret sembolü					
x		+		+	
y					
z				-	
Gerçektilen büyüklükler					
x	6.469	2.156	7.906	2.156	
y	6.469	3.594	0.000	0.000	
z	10.781	3.594	5.031	2.156	
<p>Geçiş 4</p> <p>Bit düzlemi $n = 0$</p> <p>Eşik $T_4 = 0.719$</p>	Önemlilik durumu				
	x	1	1	1	1
	y	1	1	1	1
	z	1	1	1	1
Önemli bulunan İşaret sembolü					
x					
y			-	-	
z					
Gerçektilen büyüklükler					
x	6.828	2.516	7.547	2.156	
y	6.109	3.953	1.078	1.078	
z	11.141	3.953	4.672	2.875	
<p>Geçiş 5</p> <p>Bit düzlemi $n = -1$</p> <p>Eşik $T_5 = 0.359$</p>	Önemlilik durumu				
	x	1	1	1	1
	y	1	1	1	1
	z	1	1	1	1
Önemli bulunan İşaret sembolü					
x					
y					
z					
Gerçektilen büyüklükler					
x	7.008	2.336	7.367	2.516	
y	5.930	3.773	1.258	1.258	
z	11.320	3.773	4.492	2.516	

Tablo 2 Her kodlama geçişinde işlenen bir düzlemini gösteren n değeri, katsayı büyüklüklerinin önemini sınamada kullanılan eşik değeri T_n , katsayıların önemlilik durumları, o geçişte önemli bulunan katsayıların kodlanan işaret sembolleri ve o geçiş sonrasında katsayıların gerçektilen büyüklükleri.

Tablo 2 katsayı koordinat değerlerinin inceltme kararlarını göstermese de her geçişteki inceltme kararları çıkarsanabilir. Örneğin birinci geçişte, hiçbir inceltme sembolü kodlanmamasına rağmen ikinci geçişte, ilk geçişte önemli bulunan dört katsayı için inceltme sembolleri kodlanmaktadır. Her geçiş sonunda katsayıların büyüklüklerinin gerçektilmeleri hangi inceltme kararlarının kodlandığını bize

açıklamakta kullanılabilir. Her inceltme için inceltmenin küçük veya büyük büyüklüğe doğru yapıldığı kararları kodlanmaktadır.

Kodlanan tel filenin düğümlerinin koordinat değerlerini kodçözücüde elde etmek için Tablo 3'de gösterilen, son (beşinci) geçişin sonunda elde edilen geriçatılmış spectral katsayılara bir ters dönüşüm uygulanır.

x	-7.008	2.336	-7.367	2.516
y	-5.930	-3.773	-1.258	-1.258
z	-11.320	3.773	-4.492	-2.516

Tablo 3 x,y ve z koordinatları için katsayıların geriçatımları

Tablo 3'den katsayı geriçatım matrisi \hat{C} 'yi oluşturursak Ψ matrisinin tersini kullanarak, 4x3'lük geometri matrisi G 'nin geriçatımını $G = \Psi^{-T} \hat{C}^T$ olarak elde ederiz. Geriçatılmış tel file geometrisi Şekil 8'de VRML dilindeki dosyaya kaydedilmiş olarak gösterilmektedir.

```
#VRML V2.0 utf8
Shape {
  geometry IndexedFaceSet {
    coord Coordinate {
      point [
        -2.876  1.876  1.770
         6.930 -0.922  9.023
         1.985  7.874  6.756
         7.977  3.032  5.092
      ]
    }
    coordIndex [
      0 1 2 -1
      2 3 0 -1
      2 1 3 -1
      0 3 1 -1
    ]
  }
}
```

Şekil 8 Örnek tel file geometrisinin geriçatımının VRML dilindeki dosyaya kaydedilmesi

3.5 Önemlilik testi karar sembollerinin birleşik ve basit koşullu kodlanması.

Önemlilik testlerinin karar sembollerinin birleşik veya koşullu kodlanmasıyla aralarındaki bağımlılıklar değerlendirilerek ek bir sıkıştırma kazancı sağlanabilir. Bilgi kuramının temel teoremlerinden birine göre önemlilik testlerinin birleşik entropileri, testlerin entropilerinin toplamından küçüktür. Bu matematiksel olarak

$$H(\prod_n(T_1), \prod_n(T_2)) \leq H(\prod_n(T_1)) + H(\prod_n(T_2))$$

şeklinde ifade edilebilir. Aynı eşitsizlik koşullu entropi kullanılarak

$$H(\prod_n(T_2) | \prod_n(T_1)) + H(\prod_n(T_1)) \leq H(\prod_n(T_1)) + H(\prod_n(T_2))$$

şeklinde de ifade edilebilir.

Yukarıdaki amaca yönelik iki değişik yöntem incelenmiştir. İlk yöntem iki önemlilik testi karar sembolünün birleşiminin entropi kodlanmasıdır. Karar sembol alfabesi $A = \{0,1\}$ olduğunda entropi kodlayıcının girdisi olan harf, sembol birleşim alfabesi $A \times A = \{00,01,10,11\}$ 'nin bir elemanı olmaktadır. Önemlilik testi karar sembolleri birleşik kodlanan iki küme, bölüntülenmiş önemli bir kümenin altkümeleri olduğunda,

$$\prod_n(T_1 \cup T_2) = 1 \Rightarrow \prod_n(T_1) = 1 \text{ veya } \prod_n(T_2) = 1$$

ve 00 birleşimine rastlanmaz. Bu istatistiksel bağımlılıktan dolayı sembollerin birleşik kodlanması ayrı ayrı ikili entropi kodlanmasına göre ciddi bir avantaj sağlar.

İncelenen ikinci yöntemde yukarıdaki istatistiksel bağımlılık birinci sembolün ikili entropi kodlanması, ve ikinci sembolün, birinciye koşullu olarak, ikili entropi kodlanması, veya kodlanmaması ile değerlendirilmektedir. İlk sembolün önemlilik kararı 0 olduğunda ikincinin önemlilik kararı 1 olmak zorunda olacağından kodlamaya gerek duyulmaz. Bu yöntem basit koşullu kodlama olarak adlandırılmaktadır.

Altkümelerden büyük özdeğerli katsayıları bulduranının önemsiz olma olasılığının küçük özdeğerli katsayıları bulduranının önemsiz olma olasılığına göre daha büyük olması beklenebilir. Dolayısıyla büyük özdeğerli katsayılar altkümesinin önemlilik kararının birinci sembol olarak kodlanması ile büyük olasılıkla küçük özdeğerli katsayılar altkümesinin önemliliğine belirlemci bir şekilde kodlamaya ihtiyaç duyulmadan karar verilebilir.

3.6 Önemlilik testi karar sembollerinin koşullu kodlanması ve uygun bağlamların araştırılması.

Önceki bölümde anlatılan basit koşullu kodlamaya göre kodlama kazancını daha da iyileştirmek için önemlilik testi kararlarının ve katsayı büyüklük inceltme kararlarının daha gelişmiş koşullu kodlanması da incelenmiştir. Koşullu kodlamanın yüksek sıkıştırma kazancı sağlaması için bağlam seçimi önem taşımaktadır.

Kodlanan sembol X rastgele değişkeni, kodlama bağlamı da Z rastgele değişkeni ile gösterilsin. Kodlanan sembolün entropisi $H(X)$, bağlama koşullu

entropisi de $H(X|Z)$ olduğunda $H(X|Z)$ 'yi enküçültmek $I(X;Z) = H(X) - H(X|Z)$ 'yi enbüyütmek anlamını taşımaktadır. Bu da bağlam seçiminde kodlanan sembol ile istatistiksel bağımlılığı en büyük olan bağlamın seçilmesi gerektiği anlamına gelmektedir. Özdeğerlerin büyüklüklerine göre sıralanan katsayılar için bir katsayı ile istatistiksel bağımlılığın en yüksek olduğu katsayı(lar) Şekil 3'ten görülebileceği gibi komşularıdır.

Bağlama koşullu entropinin $H(X|Z) = \sum_z H(X|Z=z) \Pr\{Z=z\}$ denklemi ile ayrıştırılabileceği görülebilir. Burada z 'nin gerçekleştirme kümesi ne kadar büyük olursa o kadar çok sayıda koşullu olasılık modeli kullanılır. Her $H(X|Z=z)$ bir koşullu olasılık modeli ile kodlama neticesinde beklenen en küçük bit hızını vermektedir. Her olasılık modelinin başlangıç durumundan son durumuna geçilmesi esnasında ortalamada $H(X|Z=z)$ 'in çok üzerinde bit harcanmaktadır. Dolayısıyla z 'nin gerçekleştirilmesinin tanım kümesinin büyük olması az sayıda düğümüne sahip küçük bir model için $H(X|Z)$ 'e erişilmesini güçleştirmektedir.

Üç koordinatın her biri için önemlilik kararlarını ve katsayı büyüklük inceltme kararlarını kodlamada değişik koşullu olasılık modelleri kullanılmıştır. Böylece x , y ve z eksenleri yönünde tel file modelinin değişik yüzey karakteristikleri yakalanabilmektedir. Örneğin *horse* modelinin bacakta bulunan bir bölgesinde z yönündeki yüzey eğriliği ortalamada x ve y yönlerindeki yüzey eğriliklerine göre daha az olduğundan z yönündeki küçük özdeğerli katsayıların daha yüksek enerjiye sahip olması beklenebilir.

Önemlilik kararları kodlanırken tek katsayı içeren ve birden fazla katsayı içeren kümeler için değişik koşullu olasılık modelleri kullanılmıştır. Kodlanan kümelerin komşuluklarına bakılarak çok sayıda standart test modeli için iyi deneysel hız-bozunum başarımı sağlayan bağlamlar saptanmıştır.

Tek katsayı veya birden fazla katsayı içeren bir kümenin en fazla iki (sol ve sağ) komşu kümesi bulunabilir. Tek katsayı kümesinin önemlilik kararı kodlanırken her iki tarafındaki en az bir komşu tek katsayı kümesinin önemlilik durumu ve varsa iki komşu tek katsayı kümesinin önemlilik durumlarının birleşimi bağlam olarak kullanılmaktadır. Benzer şekilde birden fazla katsayı içeren bir kümenin önemlilik kararı kodlanırken her iki tarafındaki en az bir, 32 katsayı içeren, komşu kümenin önemlilik durumu ve varsa iki, 32 katsayı içeren, komşu kümenin önemlilik durumlarının birleşimi bağlam olarak kullanılmaktadır. Komşu kümelerin büyüklüğü, deneysel olarak, büyük bir tel file model kümesi üzerinden, ortalama kodlama başarımını enbüyültecek şekilde saptanmıştır. Ayrıca, komşu

kümelerin önemsizlik durumlarının en son veya sondan bir önceki geçişte güncellenmiş olmasına göre farklı olasılık modelleri kullanılmaktadır.

T_R ve T_L önemlilik durum sembolü kodlanan T katsayı kümesinin komşu katsayı kümeleri olsun. T_a ($a \in \{R, L\}$) katsayı kümesinin önemlilik durumu

$$\Phi_n^s(T_a) = \begin{cases} 0 & 2^n > \max_{i \in T_a} \{ |c_{i,\gamma}| \} \\ 1 & 2^n \leq \max_{i \in T_a} \{ |c_{i,\gamma}| \} < 2^{n+1} \\ 2 & \max_{i \in T_a} \{ |c_{i,\gamma}| \} < 2^{n+1}, 2^n > \max_{i \in T_a} \{ |c_{i,\gamma}| \} \end{cases} \quad \text{olarak ifade edilsin. Tek}$$

katsayı içeren T kümesinin önemlilik sembolünün aritmetik kodlanması için kullanılan koşullu olasılık modeli $p_\gamma^1(\prod_n(T) | z)$ olsun. Benzer şekilde birden fazla katsayı içeren T kümesinin önemlilik sembolünün aritmetik kodlanması için kullanılan koşullu olasılık modeli $p_\gamma^m(\prod_n(T) | z)$ olsun. Her iki komşu da varsa $z = 3 \times \Phi_n^s(T_L) + \Phi_n^s(T_R)$, sadece T_R varsa $z = \Phi_n^s(T_R) + 9$, sadece T_L varsa $z = \Phi_n^s(T_L) + 9$ bağlam olarak alınır.

Katsayı büyüklük inceltme bağlamları inceltilen katsayının ve en yakın bir veya varsa iki komşusunun inceltme durumları kullanılarak belirlenir. İlk kez inceltilen katsayılar ve daha önceden inceltilmiş katsayılar değişik olasılık modelleri kullanılarak kodlanırlar. Daha önceden inceltilmiş katsayılar arasında en az bir önceden inceltilmiş komşusu bulunanlar ve hiçbir inceltilmiş komşusu bulunmayanlar da değişik olasılık modelleri kullanılarak kodlanırlar.

$c_{i,\gamma}$ katsayısının b inceltme sembolünün aritmetik kodlanması için kullanılan koşullu olasılık modeli $p_\gamma^r(b | z)$ olsun. $c_{i,\gamma}$ katsayısının inceltme durumu olan

$$z = \begin{cases} 0 & 2^n > |c_{i,\gamma}| \\ 1 & 2^n \leq |c_{i,\gamma}| \text{ ve } (2^n \leq |c_{L,\gamma}| \text{ veya } 2^n \leq |c_{R,\gamma}|) \\ 2 & 2^n \leq |c_{i,\gamma}| \text{ ve } 2^n > |c_{L,\gamma}| \text{ ve } 2^n > |c_{R,\gamma}| \end{cases}$$

bağlam olarak alınır.

3.7 Değişmez taban vektörleri ile verimli geometri kodlama için çoktan teke eşlemler

Bölüm 2.4'de anlatılan (KARNI, 2001)'de önerilmiş olan $M(\cdot)$ birebir eşlemleri dönüşüm öncesinde her düzenli tel file düğümüne karşılık gelen en uygun kodlanan düzensiz tel file düğümünü belirtmekte ve $M^{-1}(\cdot)$ de ters dönüşüm sonrasında düzenli tel file düğümleri için geriçatımlar yapıldıktan sonra her düzensiz tel file düğümüne karşılık gelen en uygun düzenli tel file düğümünü belirtmektedir. Birebir eşlemlerin saptanmasında düzenli ve düzensiz tel file düğüm bölgelerinin birim dairelerde noktalar ile temsil edilmesi ve iki birim dairede Euklid uzaklığını enküçülten birer noktanın eşlemlenmesi amaçlanmaktadır. İdealde bu amaca ulaşılsa bile birebir eşlemlerinin kullanılması eniyilikten uzaktır. Bunun nedenini basit bir örnek ile izah edebiliriz.

Örneğimizde düzenli tel file bölgesindeki bir düğümü temsil eden birim dairedeki p noktasının düzensiz tel file bölgesindeki iki düğümü temsil eden birim dairedeki q_1 ve q_2 noktalarına olan Euklid uzaklıkları en küçük ve yaklaşık olarak aynı olsun. Bu durumda bile birebir eşleme q_1 ve q_2 'den yalnızca birini (diyelim ki q_1) tercih edip ikincisini gözardı edecektir. Sadece q_1 'in temsil ettiği düğümün koordinatları düzenli tel file bölgesinde eşlemlenen düğümün koordinatları olarak dönüşüme sokulacaktır. Halbuki Euklid uzaklığı yaklaşık olarak aynı olan q_2 'nin de düzenli tel file bölgesindeki düğüm koordinatlarının belirlenmesinde eşit oranda katkısı olması beklenebilir. Sadece q_1 'i seçip temsil ettiği düğümün koordinatlarını kullanarak p 'nin temsil ettiği düğümün koordinatlarını belirlemek yerine eşit Euklid uzaklığına sahip q_1 ve q_2 'nin ikisini beraber seçip bunların temsil ettikleri düğümlerin koordinatlarının ortalamasını p 'nin temsil ettiği düğümün koordinatlarını belirlemek için kullanmak daha anlamlı olacaktır.

Yukarıda verilen örnekteki eşleme çoktan bire tipte olup genelleştirilebilir. Düzenli tel file bölgesindeki \bar{s}_j düğümünü birim dairede p_j noktası temsil ediyor olsun. p_j 'ye düzensiz tel file bölgesini temsil eden birim dairedeki en yakın K nokta $q_{M(j,1)}, \dots, q_{M(j,K)}$ olarak saptansın. Burada $M(j,k)$ dönüşüm öncesinde kullanılan birinci çoktan bire eşlemleri göstermektedir. En küçük K

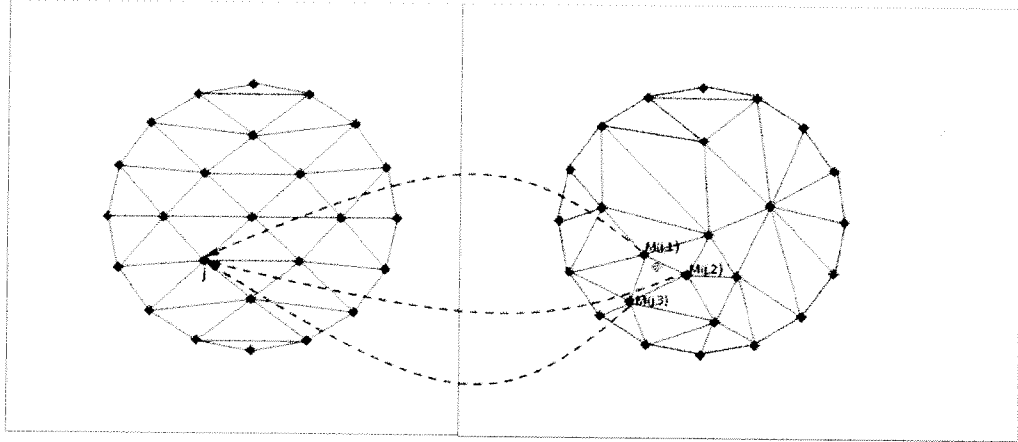
Euklid uzaklığı $\delta_{j,k} = \|p_j - q_{M(j,k)}\|_2$ ve bunlara bağlı ağırlıklar

$$w_{j,k} = \begin{cases} \frac{\delta_{j,k}^{-1}}{\sum_{k=1}^K \delta_{j,k}^{-1}} & \delta_{j,k} \neq 0 \\ 1 & \delta_{j,k} = 0 \end{cases} \quad \text{olarak hesaplanabilir. Düzenli tel file bölgesindeki}$$

düğümün koordinatlarından oluşan vektörün j 'nci bileşeni düzensiz tel file bölgesinde eşlemlenen K düğümün koordinatlarının ağırlıklı ortalaması olarak

$$\bar{s}_{\gamma,j} = \sum_{k=1}^K w_{j,k} S_{\gamma,M(j,k)} \quad (3)$$

şeklinde hesaplanır. Şekil 9'da dönüşüm öncesinde düzensiz birim dairedeki noktaların düzenli birim dairedeki eşlemlenmesi bir örnek yardımıyla anlatılmaktadır.



Şekil 9 Dönüşüm öncesinde düzensiz tel file bölgesini temsil eden birim dairedeki noktaların (sağda) düzenli tel file bölgesini temsil eden birim dairedeki noktalara (solda) çoktan teke eşlemlenmesi. Bu örnekte $N=5$ olup $K=3$ alınmıştır. Sağdaki birim dairedeki üç noktanın soldaki birim dairede eşlemlendiği noktanın konumu sağdaki birim daire üzerinde açık renkle belirtilmiştir.

Benzer bir şekilde CSPECK kodlama ve ters dönüşüm sonrasında elde edilen düzenli tel file bölgesindeki düğüm koordinatlarının geri çatımları, ikinci bir eşleme aracılığıyla düzensiz tel file bölgesindeki düğüm koordinatlarının geriçatımlarını saptamada kullanılırlar.

Düzensiz tel file bölgesindeki s_j düğümünü birim dairede q_j noktası temsil ediyor olsun. q_j 'e düzenli tel file bölgesini temsil eden birim dairedeki en yakın K nokta $p_{\tilde{M}(j,1)}, \dots, p_{\tilde{M}(j,K)}$ olarak saptansın. Burada $\tilde{M}(j,k)$ ters dönüşüm

sonrasında kullanılan ikinci çoktan bire eşlelemeyi göstermektedir. En yakın K Euklid uzaklığı $\varepsilon_{j,k} = \|q_j - p_{\tilde{M}(j,k)}\|_2$ ve bunlara bağlı ağırlıklar

$$\tilde{w}_{j,k} = \begin{cases} \frac{\varepsilon_{j,k}^{-1}}{\sum_{k=1}^K \varepsilon_{j,k}^{-1}} & \varepsilon_{j,k} \neq 0 \\ 1 & \varepsilon_{j,k} = 0 \end{cases} \text{ olarak hesaplanabilir. Düzensiz tel file bölgesindeki}$$

düğümün geriçatılmış koordinatlarından oluşan vektörün j 'nci bileşeni düzenli tel file bölgesinde eşlemlenen K düğümün geriçatılmış koordinatlarının ortalaması olarak

$$\hat{s}_{\gamma,j} = \sum_{k=1}^K \tilde{w}_{j,k} \hat{s}_{\gamma,\tilde{M}(j,k)} \quad (4)$$

şeklinde hesaplanır.

Yapılan eşlemlerde eğer j düzenli (düzensiz) tel file bölgesinin sınır düğümüne ait bir indis ise $M(j,k)$ 'nin ($\tilde{M}(j,k)$ 'nin) de düzensiz(düzenli) tel file bölgesinin sınır düğümüne ait bir indis olması ve ayrıca $K = 2$ olması koşulları uygulanmaktadır. Böylelikle düzenli(düzensiz) tel file sınır düğümleri sadece düzensiz(düzenli) tel file sınır düğümleri cinsinden ifade edilmektedir.

Bu çalışmada önerilen yöntemde ilk olarak çeşitli büyüklükte ($N=5..30$) düzenli altıgen tel file bölgeleri için taban vektör kümeleri tasarlanmaktadır. Kodlanacak düzensiz tel file bölgesinin toplam düğüm sayısı bulunmakta, toplam düğüm sayısı bu sayıdan büyük olan en küçük düzenli tel file bölgesi saptanmakta ve bu tel file bölgesi için tasarlanmış olan taban vektörleri dönüşümde kullanılmaktadır. Bu çalışmada önerilen yöntemde uygulanan eşlemler Bölüm 2.4'te anlatılan yöntemde olduğu gibi bire bir olmadığından eşlemlenen bölgelerdeki düğüm sayılarını eşitlemek için düğümlerin ilave edilmesine gerek görülmemiştir. Bu da hesaplama karmaşıklığında az da olsa bir azalma sağlamaktadır.

Denklem 3 ve 4'te uygulanan aradeğerleme işlemleri alçak geçiren süzgeçleme işlevi gördüğünden tel file geometrisindeki ayırıt, çıkıntı gibi yüksek frekans bilgileri bu işlemler neticesinde kaybedilmektedir. Bölgeye uyarlamalı dönüşüm ve CSPECK ile kodlamada düşük bit hızlarında genelde düşük frekans bilgileri kodlanmakta, yüksek bit hızlarına çıkıldığında ise yüksek frekans bilgileri kodlanmaya başlanmaktadır. Yukarıda anlatılan değişmez taban vektörleri ile yapılan dönüşüm ve CSPECK ile kodlamada düşük bit hızlarında düşük frekans bilgileri kodlanırken kodlama başarımında eniyilikten fazla bir sapma olmamakla beraber yüksek bit hızlarına çıkılınca kaybedilen yüksek frekans bilgileri yerine

düşük frekans bilgilerinin kodlanmaya devam edilmesi başarım kaybına neden olmaktadır.

Bu probleme çözüm olarak her tel file bölgesi için kodlama başarımının izlenmesi ve herhangi bir CSPECK geçişinde bozunum kazanımının o geçiş ve sonrasında belli bir seviyenin altında kalacağı öngörüldüğünde dönüşüm tabanlı kodlamanın o bölge için sonlandırılıp o bölgedeki koordinat hatalarının uzamsal olarak kodlanması önerilmektedir.

CSPECK kodlama öncesinde katsayıların büyüklükleri bilindiği için her geçiş için düzenli tel file bölgesi düğümlerinin koordinatlarının geriçatımları ve eşleme aracılığıyla da düzensiz tel file bölgesi düğümlerinin koordinatlarının geriçatımları bulunabilir. Geçiş n ve $n-1$ arasında bozunumda elde edilen ortalama azalma $\Delta D_n = D_n - D_{n-1}$ olarak ifade edilsin. Eğer $\Delta D_k < 0.3D_k$ $k = n, n-1, \dots$ gözlemlenirse tel file bölgesinin dönüşüm tabanlı kodlanması n değeri için olan geçişte sonlandırılmaktadır. Bu durumda yüksek frekans bilgisinin kaybedilmesi $n-1$ ve daha küçük değerler için olan geçişlerde bozunumda yeterli kazanımın (azalmanın) elde edilememesine neden olmaktadır.

Bütün kodlama sisteminin çalıştırılmasında kullanıcı arzuladığı bir karesel hata bozunum değeri olan d_{istek} girdiler. Eğer herhangi bir tel file bölgesi için $D_n > d_{istek}$ ise o bölgedeki düğüm koordinat hataları CSPECK kodlamanın tamamlanmasından sonra ardışık yaklaşıklama sayıl nicemleyici ile uzamsal olarak kodlanır.

Uzamsal kodlama öncesinde kodlanacak bütün düğüm koordinat hatalarının en büyük genliği bulunur ve bu değer 32 bit olarak kodçözücü tarafa iletilir. Uzamsal kodlamada kullanılacak ilk eşik değeri δ bu değer yarısı olarak alınır. Uzamsal kodlama esnasında her geçişte bir önceki geçişte kullanılan eşik değerinin yarısı yeni eşik değeri olarak kullanılır. Her geçişte daha önceki geçişlerde önemsiz olarak saptanmış (büyüklükleri önceki geçişlerde kullanılan eşik değerlerinden küçük) koordinat hataları için önemlilik kararları kodçözücüye entropi kodlanır. Önemli olarak saptanan katsayıların işaretleri ikili olarak kodlanır. İlk kez önemli olarak saptanan koordinat hatalarının ilk geriçatımları $\pm 1.5\delta$ olarak gerçekleştirilir. Diğer bir deyişle düğüm koordinatlarının geriçatımlarına $\pm 1.5\delta$ eklenir. Ayrıca her geçişte daha önceki geçişlerde önemli olarak bulunmuş koordinat hatalarının inceltmesi yapılır. Kalan hatanın işareti ikili kodlanırken düğüm koordinatlarının geriçatımları kalan hatanın işaretine göre $\pm 0.5\delta$ eklenerek güncellenir.

Her tel file bölgesi için uzamsal kodlamada kullanılan geçiş sayısı 6 bit ile ikili olarak kodlanarak kodçözücüye iletilmektedir. Ayrıca uzamsal kodlamanın yapıp yapılmadığı 1 bit ile kodçözücüye bildirilmektedir.

4. Bulgular ve Tartışma

Bu bölümde deneylerde elde edilen sonuçlar iki grupta toplanarak sunulmaktadır. İlk grupta Bölüm 2.2'de anlatılan uyarlamalı spektral dönüşümün Bölüm 3.1-3.6'da gelişimi anlatılan ve önerilen CSPECK kodlayıcıyla bütünleştirilmesi ile elde edilen sonuçlar sunulmaktadır. Ayrıca Bölüm 2.3'de sunulan spektral tel file sıkıştırma yöntemi ile elde edilen sonuçlar, (TOUMA, 1998)'deki öngörücü kodlama yöntemi ile elde edilen sonuçlar ve dalgacık tabanlı kodlama yöntemi ile elde edilen kodlama sonuçları karşılaştırma için sunulmaktadır.

İkinci grupta ise Bölüm 3.7'de önerilen değişmez taban vektörleri ile dönüşüm yönteminin kullanıldığı CSPECK kodlayıcı ile elde edilen kodlama sonuçları sunulmaktadır. Ayrıca karşılaştırma için Bölüm 2.4'te anlatılan değişmez taban vektörleri ile dönüşüm yönteminin (KARNI, 2001) kullanıldığı CSPECK kodlayıcı ile elde edilen kodlama sonuçları ve uyarlamalı dönüşüm yönteminin kullanıldığı CSPECK kodlayıcı ile elde edilen kodlama sonuçları verilmektedir.

Benzetimlerde 19,851 düğüm ve 39,698 yüzden oluşan *horse*, 34,835 düğüm ve 69,472 yüzden oluşan *bunny*, ve 50002 düğüm ve 100000 yüzden oluşan *venus* modelleri kullanılmıştır. Uyarlamalı dönüşümün kullanıldığı kodlama sistemlerinde kodlama kazancını yüksek ve hesaplama karmaşıklığını makul düzeyde tutmak üzere her tel file bölgesindeki ortalama düğüm sayısı ~ 500 olmak üzere *horse* 40, *bunny* 70 ve *venus* 100 bölgeye bölüntülenmiştir.

Değişmez taban vektörleri ile dönüşüm yönteminde düzensiz tel file bölgesindeki düğümlerin birim dairedeki noktalar ile iyi bir şekilde temsil edilebilmesi için bölgenin basitçe bağlantılı olması gerekmektedir. Çünkü dönüşüm için kullanılan düzenli tel file bölgesi basitçe bağlantılıdır. Öte yandan MeTIS yazılımının uygulanması bazen basitçe bağlantılı olmayan tel file bölgeleri vermektedir. Bu durumda basitçe bağlantılı olmayan her tel file bölgesi MeTIS'in özyineli olarak uygulanması ile basitçe bağlantılı olmayan tel file bölgeleri kalmayınca kadar ikiye bölüntülenmektedir. Örneğin *horse* modeli MeTIS'in ilk uygulanması ile 40 tel file bölgesine ayrıldığında bacak kesimlerindeki tel file bölgelerinden bazıları basitçe bağlantılı olmadığından ilave bölüntüleme işlemleri uygulanmakta ve sonrasında hepsi basitçe bağlantılı olan 70 tel file bölgesi elde edilmektedir. Bu durumda küçük tel file bölgelerinin bazıları 25, 36 gibi az sayıda düğüme sahip olmaktadır.

Özdeğer ve özvektör hesaplama işleminin, hızlı çalışan, yeterli hesaplama kesinliğine sahip, ve geliştirilmekte olan kodlayıcı sisteminin diğer bileşenleri ile

kolaylıkla bütünleştirilebilen bir matematik yazılım paketi tarafından yapılmasının gerekliliği görülmüştür. Bu amaç doğrultusunda CLAPACK (C Linear Algebra Packet) ücretsiz yazılım paketinin MATLAB gibi bir ticari pakete göre hesaplama kesinliği, C kaynak kodu ile bütünleştirme kolaylığı ve hızlı çalışması gözetilerek kullanılmasına karar verilmiştir.

4.1 Ölçevler

Özgün bir model M ile geriçatılmış \hat{M} 'nin arasındaki farkı ve ortaya çıkan kaybı belirlemede ortalama karesel hata cinsinden bir ölçüm yapmak, geometrik yakınlığın bir göstergesini verse de insan gözünün algıladığı düzlük gibi önemli görsel özellikleri yakalayamadığından doğru bir hata ölçevi değildir. (KARNI, 2000)'deki görsel hata ölçevi, düğümler arasındaki Euklid uzaklığıyla birlikte düğümlerdeki düzlük değerleri arasındaki farkı da gözettilmesi için tercih edilmiştir.

İndisi i olan düğümün komşu düğümler kümesini $n(i)$, ve indisleri i ve j olan düğümler arasındaki geometrik (Euklid) uzaklığı l_{ij} ile gösterelim. Geometrik Laplas işleci,

$$GL(v_i) = v_i - \frac{\sum_{j \in n(i)} l_{ij}^{-1} v_j}{\sum_{j \in n(i)} l_{ij}^{-1}} \quad (4)$$

düğüm v_i 'deki düzlük vektörünü verir. V ve \hat{V} , M ve \hat{M} modellerine ait düğüm kümeleri olsun. Bu modeller arasındaki görsel uzaklık, düğümler arasındaki geometrik uzaklıkla düzlük vektörleri arasındaki uzaklığın ortalaması olarak

$$D(M, \hat{M}) = \frac{1}{2N} (\|V - \hat{V}\|_2 + \|GL(V) - GL(\hat{V})\|_2) \quad (5)$$

denklemlerle ifade edilebilir. Burada $\|V - \hat{V}\|_2 = \sqrt{\sum_{v_i \in V} \|v_i - \hat{v}_i\|_2^2}$ olarak tanımlanmaktadır.

Düğüm koordinat yaklaşılama kalitesini gösteren görsel hata ölçevinin aksine Hausdorff ve etkin değer uzaklıkları, modelin özgün yüzeyinin kodlama neticesinde geriçatılmış yüzey ile yaklaşılmasının kalitesini göstermektedirler. Bu ölçevler yeniden örme (remeshing) gibi topolojide değişikliğe neden olan önışlemler sonrasında tel file geometrisinin sıkıştırma başarımını ölçmede kullanılmaktadırlar. Yüzeyler S ve S' arasındaki simetrik Hausdorff ve etkin değer uzaklıkları sırasıyla

$$d_h(S, S') = \max(\max_{p \in S} d(p, S'), \max_{p \in S'} d(p, S))$$

ve

$$d_{mse}(S, \hat{S}) = \left(\frac{1}{|S|} \iint_{p \in S} d(p, \hat{S})^2 dS \right)^{1/2}$$

olarak tanımlanmaktadır. Burada $d(p, S) = \min_{p \in S} \|p - p'\|$ özgün yüzeydeki bir noktanın geriçatılmış yüzeye olan uzaklığıdır.

4.2 Uyarlamalı dönüşüm yönteminin kullanıldığı önerilen CSPECK kodlayıcı ile elde edilen sonuçlar

Şekil 10'da uyarlamalı dönüşüm yönteminin kullanıldığı CSPECK kodlayıcısının Bölüm 3.5'te anlatılan birleşik ve basit koşullu sembol kodlama yöntemleri ve Bölüm 3.6'da anlatılan gelişmiş koşullu sembol kodlama yöntemi ile gerçekleştirmelerinin karşılaştırmalı hız-bozunum başarımları gösterilmektedir. Karşılaştırma için Bölüm 2.3'deki spektral sıkıştırma yöntemi (KARNI, 2000) 12, 14 ve 16 bit nicemleme çözünürlükleriyle ve (TOUMA, 1998)'deki öngörücü kodlama yöntemi gerçekleştirilmiştir.

Katsayılar vektörünü kırparak yüksek enerjili altvektörü kodlamaya dayanan spektral sıkıştırma yönteminin hız bozunum başarımının düşük bit hızlarında çok iyi olduğu gözlemlenmektedir. Buna rağmen bölge ve koordinatlara bit ataması yapılmadığından ve katsayıların bitlerine doğru öncelik verilmediğinden, hız-bozunum başarımı, orta ve yüksek bit hızlarında öngörüye dayanan yöntemlerinkine göre düşük kalmaktadır. Önerilen yöntem, bölge ve koordinatlara dolaylı bit ataması yaptığından, katsayıların bitlerine doğru önceliği verdiğinden, ve birden fazla önemsiz katsayının sıfır bitini tek sembolle birleşik kodladığından orta ve yüksek hızlarda da iyi bir başarımla sağlamaktadır.

Kodlayıcının gerçekleştirilmesinde ebeveyn kümesi önemli bulunan altkümelerin önemlilik sembolleri birleşik veya koşullu kodlandığı takdirde bit hızında bağımsız kodlamaya göre %5-%10 azalma gözlemlenmektedir. Basit koşullu kodlama ve sembol birleşiminin entropi kodlanması yöntemleri beraberlerinde çok az hesaplama karmaşıklığı artışları getirdikleri için bu yöntemlerden birini önerdiğimiz kodlama sisteminin temel bir bileşeni olarak öneriyoruz. Şekil 11'de horse modeli için 0.22, 0.81, 7.12 ve 14.34 bit/düğüm hızlarında sembol birleşiminin entropi kodlanması ile elde edilen geriçatım sonuçları görselleştirilmektedir.

Genelde basit koşullu kodlama ve sembol birleşiminin entropi kodlanması yöntemleri birbirine çok yakın hız-bozunum başarımı sağlamıştır. Bununla

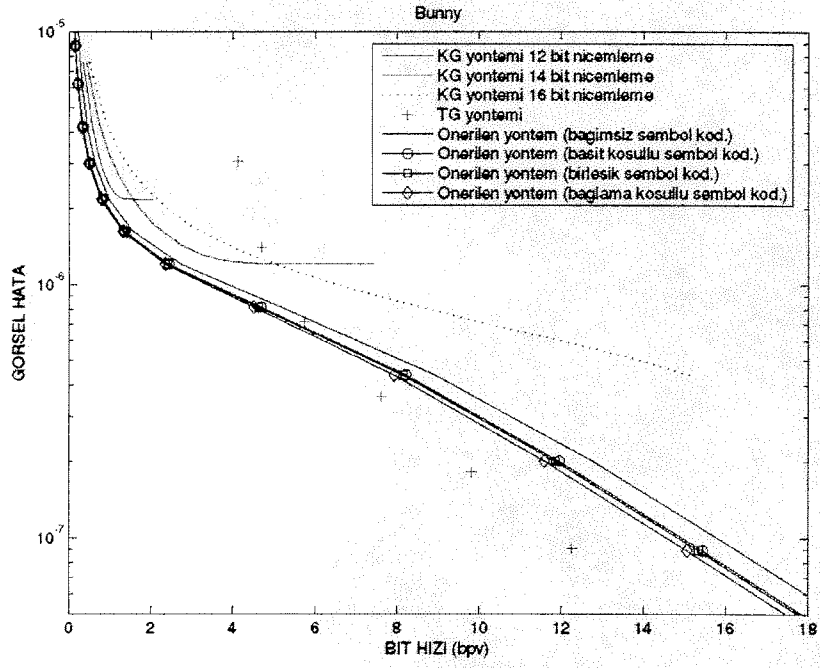
beraber Bölüm 3.6'da anlatılan bağlama koşullu sembol kodlama yöntemi orta ve yüksek bit hızlarında aynı bozunum için 0.2-0.4bit/düğüm gibi nisbeten az ama yoksanamayacak bir hız avantajı sağlamaktadır. Bu avantajın sınırlı kalması ve daha fazla hız bozunum avantajı sağlayacak bağlamaların saptanamamış olmasının sebebinin katsayılar vektöründe yapılan sıralamanın özdeğerlere göre belirlenmesi olduğu olasıdır. Dalgacık tabanlı görüntü kodlamada olduğu gibi komşu katsayılar uzamsal olarak ilişkili olmadığından bunların büyüklükleri arasında da kodlama sisteminin değerlendirebileceği çok kuvvetli ilintiler bulunmamaktadır.

Tablo 4'den görülebileceği gibi önerilen kodlayıcı (KHODAKOVSKY, 2002)'de önerilen dalgacık tabanlı Dik Tel file Sıkıştırma yöntemine özellikle orta ve yüksek bit hızlarında iyi bir alternatif sunmaktadır. (KHODAKOVSKY, 2002)'de kullanılan yüksek frekans bilgisinin kaybına sebebiyet veren yeniden örme önişlemi yüksek bit hızlarında göz ardı edilemeyecek bozunuma sebebiyet vermektedir.

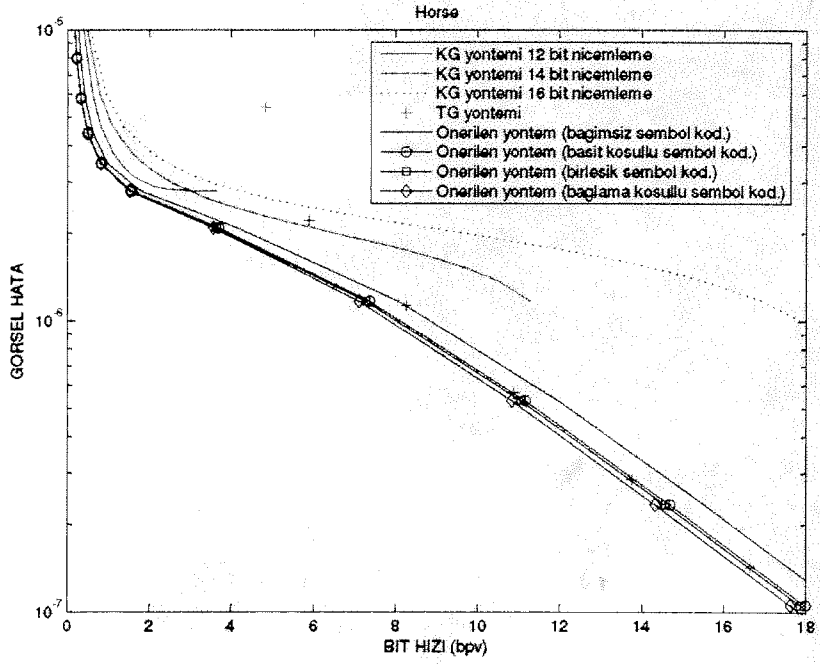
Şekil 12'ye bakıldığında önerilen uyarlamalı dönüşüm yönteminin kullanıldığı CSPECK kodlayıcının spektral sıkıştırma yöntemine göre özgün modelin daha iyi bir geriçatımını verdiği görülmektedir. Her ne kadar (KHODAKOVSKY, 2002)'de önerilen dalgacık tabanlı yöntem önerilen kodlama yöntemine göre düz yüzey bölgelerinin özgün modeldekine daha benzer geriçatımlarını verse de önerilen kodlama yöntemi yüksek frekans bölgelerinin daha iyi bir şekilde korunmasını sağlamaktadır.

Önerilen kodlayıcı			Dik Tel file Sıkıştırma (KHODAKOVSKY, 2002)		
Hız	Hausdorff (E-4)	Etkin değer (E-5)	Hız	Hausdorff (E-4)	Etkin değer (E-5)
11.60	4.81	1.65	11.554	53.50	2.45
4.51	18.64	6.25	4.665	53.70	3.7
2.34	33.57	9.95	2.368	53.32	6.3
1.31	49.2	15.7	1.220	74.20	10.6
0.79	69.16	24.1			

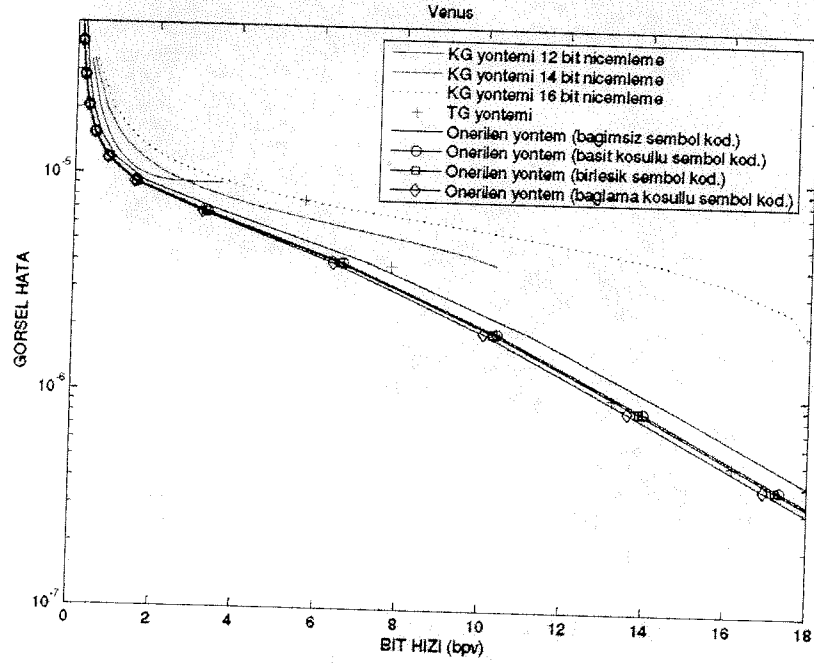
Tablo 4 Bunny modeli için uyarlamalı dönüşüm yönteminin kullanıldığı önerilen CSPECK kodlayıcı (bağlama koşullu sembol kodlama yöntemi) ve dalgacık tabanlı kodlayıcı (KHODAKOVSKY, 2002) için Hausdorff (kuşatan kutu diyagonaline göre) ve etkin değer uzaklıklarının hıza göre değişimi.



a)

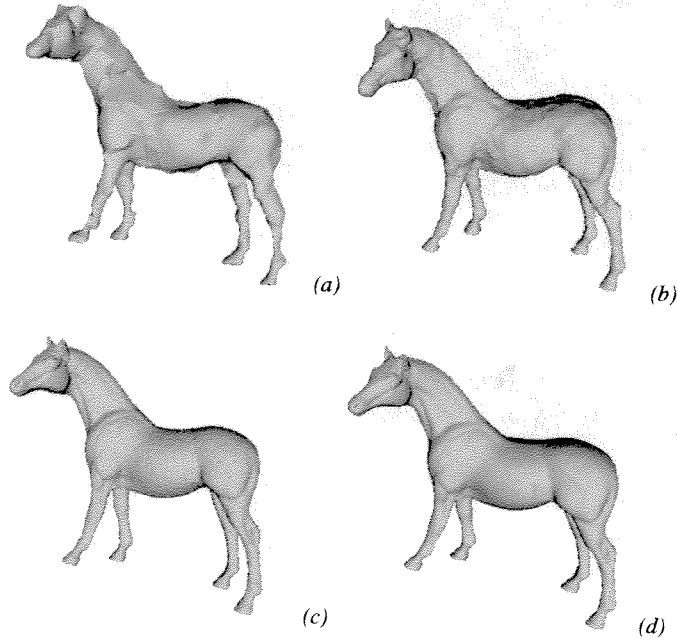


b)

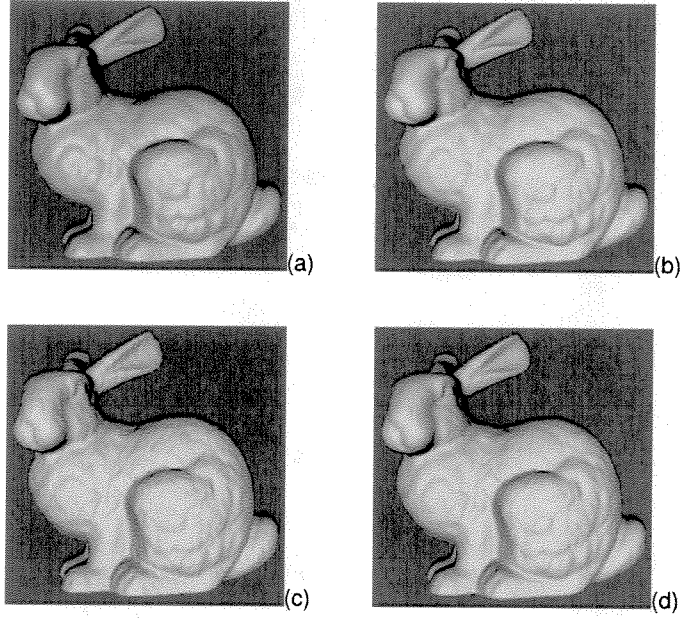


c)

Şekil 10 Uyarlamalı dönüşüm yönteminin kullanıldığı önerilen CSPECK kodlayıcının sınanması için benzetim sonuçları a) Bunny, b) Horse ve c) Venus tel fileleri



Şekil 11 Horse modelinin uyarlamalı dönüşüm yönteminin kullanıldığı önerilen CSPECK kodlayıcı ile aşamalı geriçatımları (a)0.22 bpv, (b)0.81 bpv, (c)7.12 bpv, (d)14.34 bpv



Şekil 12 Bunny modeli a): Özgün model, b) ve c): 4.67 bit/düğüm bit hızında sırasıyla dik tel file sıkıştırma yöntemi (KHODAKOVSKY, 2002) ve spektral sıkıştırma yöntemi (KARNI, 2000) ile geriçatımlar d): 4.51bit/düğüm bit hızında uyarlamalı dönüşüm yönteminin kullanıldığı önerilen CSPECK kodlayıcı ile geriçatım.

4.3 Önerilen değişmez taban vektörler ile dönüşüm yöntemi ile elde edilen sonuçlar

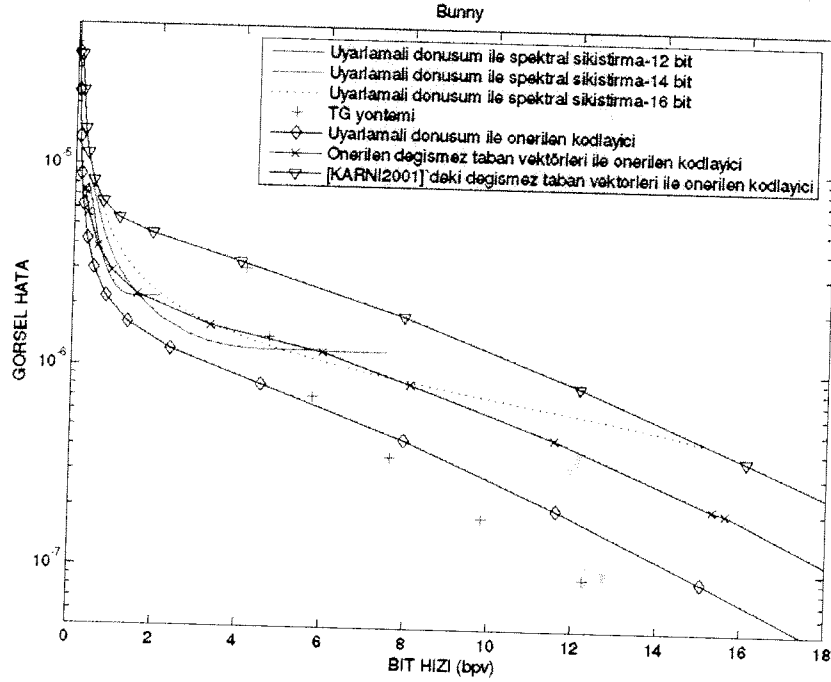
Şekil 13'de Bölüm 3.7'de önerilen çoktan teke eşlemlerinin kullanıldığı değişmez taban vektörler ile dönüşüm yönteminin kullanıldığı CSPECK kodlayıcısı ile elde edilen hız-bozunum başarımları sunulmaktadır. Karşılaştırma için önceki sonuçların yanında Bölüm 2.4'te anlatılan (KARNI, 2001)'de önerilmiş olan bire bir eşlemlerinin kullanıldığı değişmez taban vektörler ile dönüşüm yönteminin kullanıldığı CSPECK kodlayıcısı ile elde edilen hız-bozunum başarımları da sunulmaktadır.

(KARNI, 2001)'de önerilen değişmez taban vektörler ile dönüşüm yöntemi uyarlamalı dönüşüm yönteminden oldukça düşük başarımlar göstermektedir. Başarımdaki kayıp yüksek bit hızlarına çıktığında (15bit/düğüm>) bir büyüklük mertebesini aşmakta ve hesaplama karmaşıklığında sağlanan kazanım (azalma) başarımdaki kayba göre yetersiz kalmaktadır. Diğer taraftan bu çalışmada önerilen yöntem özellikle düşük bit hızlarında hız-bozunum başarımlarından

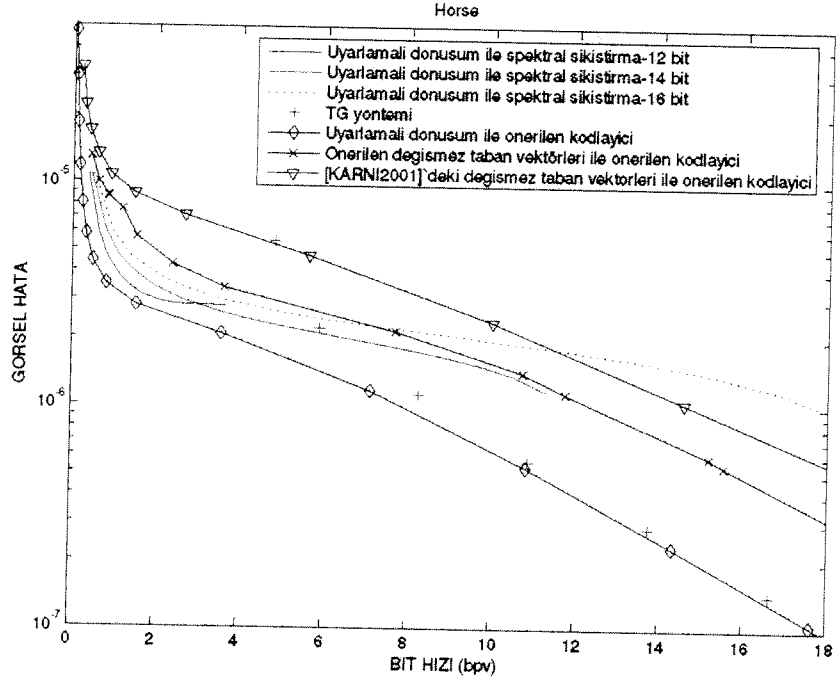
göreceli olarak daha az ödün vererek hesaplama karmaşıklığında benzer bir kazanım sağlamaktadır.

Değişmez taban vektörler ile dönüşüm yöntemlerinin uygulanmasında her bölge için her defasında dönüşüm hesaplanmamakta, NXN'lik ($N=5..30$) düzenli tel file bölgeleri için önceden hesaplanmış olan dönüşüm taban vektörleri hafıza ya da sürücüde tutulup gerektiğinde getirilmektedir. Bu da hafıza gereksinimi veya sürücü giriş-çıkış (I/O) yükünde uyarlamalı dönüşüm yöntemine göre ciddi bir artışa neden olmaktadır.

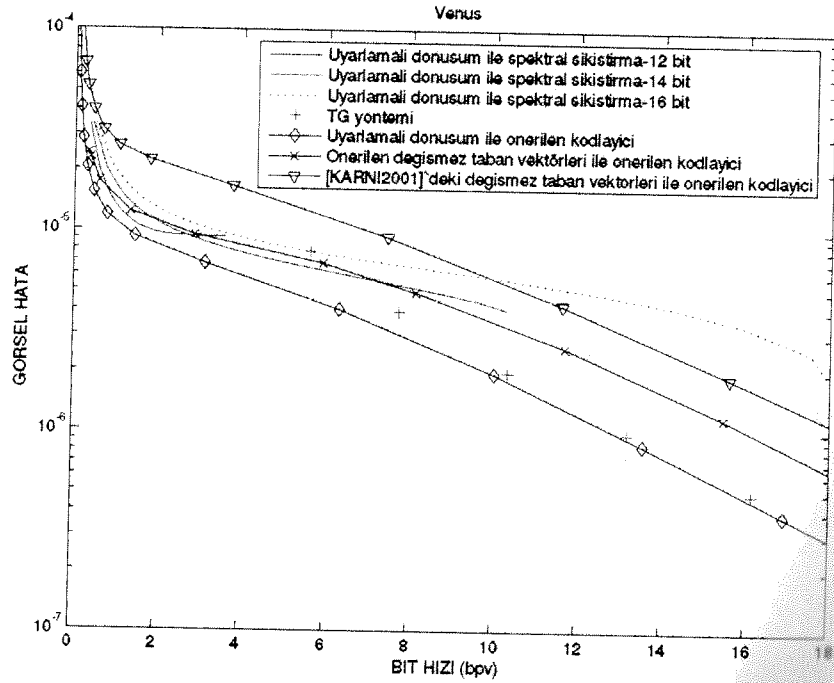
Düşük hesaplama karmaşıklığına sahip öngörüye dayalı (TOUMA, 1998)'deki yöntem ile karşılaştırıldığında 4bit/düğüm hızının altında önerilen kodlama sisteminin başarımlı avantajı sağladığı görülmektedir. 4bit/düğüm hızının üstünde ise (TOUMA, 1998)'de önerilen yöntemin avantajlı olduğu görülmektedir. Bununla beraber çalışmamızda önerilen yöntemin aksine (TOUMA, 1998)'de önerilen yöntemin bir aşamalı kodlama yöntemi olmayıp kanal hatalarına daha hassas olduğu unutulmamalıdır.



a)



b)



c)

Şekil 13 Önerilen değişmez taban vektörleri ile dönüşüm yönteminin sınavması için yapılan benzetimlerin sonuçları: a) Bunny, b) Horse ve c) Venus tel fileleri

4.4 Hesaplama karmaşıklığı ölçümleri

Uyarlamalı dönüşüm ve önerilen değişmez taban vektörler ile dönüşüm yöntemlerinin kullanıldığı CSPECK kodlama yazılım sistemlerinin yürütme hızlarını saptamak için bazı ölçümler yapılmıştır. Ölçümler Intel Core2 T7200 işlemcilerine, 2.00 GB hafızaya ve Windows XP Media Center Edn. Service Pack 2 sahip bir dizüstü bilgisayarda gerçekleştirilmiştir. Ölçümler aritmetik kodlama ve tel filenin yüzey bölüntülenmesi için gerekli zamanları da içermektedir.

Giriş-çıkış için harcanan zamanlar ölçülmediğinde 50002 düğümlü Venus modelinin geometrisinin önerilen değişmez taban vektörler ile dönüşüm yönteminin kullanıldığı CSPECK kodlayıcısı ile kodlanmasının ve kodçözümünün beş yürütme üzerinden ortalama yürütme sürelerinin sırasıyla 34.79 saniye ve 16.76 saniye olduğu saptanmıştır. Bu ölçümler 19bit/düğüm bit hızında elde edilmiştir.

Bununla beraber yine giriş çıkış için harcanan zaman ölçülmediğinde aynı modelin uyarlamalı dönüşüm yönteminin kullanıldığı CSPECK kodlayıcısı ile kodlanmasının beş yürütme üzerinden ortalama yürütme süresinin 927.47 saniye olduğu saptanmıştır. Elde edilen sonuçlar önerilen değişmez taban vektörleri yönteminin hesaplama karmaşıklığı avantajını doğrular niteliktedir.

Yürütme sürelerinin ölçümleri için kullanılan kodlayıcı ve kodçözücü yazılımların hıza yönelik eniyileme çalışmalarının ilerletilmesi ile yukarıda verilen sürelerin düşürülebileceğine inanılmaktadır.

5. Sonuç

Bu projedeki çalışmalar esnasında öncelikle (KARNI, 2000)'de önerilen uyarlamalı dönüşüm yöntemi ile görüntü kodlama için geliştirilmiş olan CSPECK küme bölüntüleme kodlama yöntemi başarı ile bütünleştirilmiştir. Geliştirilen yeni tel file geometri sıkıştırma sistemi başarımlı açısından (KARNI, 2000)'de önerilen sıkıştırma yöntemine göre bütün bit hızlarında, (KHODAKOVSKY, 2002) tarafından önerilen dalgalık dönüşüm tabanlı yöntemine göre de yüksek bit hızlarında avantaj sağlamaktadır. Ne var ki uyarlamalı dönüşüm için gerekli hesaplama yükü birçok uygulama için sistemin bu bileşeninin gerçekleştirilmesini güçleştirmektedir.

Düşük hesaplama karmaşıklığına sahip bir çözüm olarak (KARNI, 2001)'de önerilmiş olan değişmez taban vektörleri ile dönüşüm yöntemi ise başarımlı açısından tatminkar olmaktan uzaktır. Çalışmamızda aynı amaç doğrultusunda geliştirdiğimiz ve önerdiğimiz değişmez taban vektörleri ile dönüşüm yöntemi ise hesaplama karmaşıklığında benzer bir azalma sağlamakla kalmayıp başarımlı uyarlamalı dönüşüme göre olan kaybı azaltmaktadır. Önerilen yöntemin en kritik noktalarından biri de tel file bölgesi bazında dönüşüm tabanlı kodlamadan uzamsal kodlamaya geçiliyor olmasıdır. Her tel file bölgesi için hangi kodlama aşamasında dönüşüm tabanlı kodlamanın sonlandırılıp uzamsal kodlamanın başlatılacağı ise araştırma gerektiren iyileştirmeye açık bir konudur.

Proje çalışmalarının son evresinde düşündüğümüz ve yakın gelecekte incelemeyi planladığımız bir başka konu da bir sahnede yer alan birden çok tel file modelinin birlikte kodlanmasıdır. Birlikte kodlamadaki en önemli kodlama problemi tel file modellerine bit atanmasıdır. Tel file modellerini ayrı ayrı kodlayıp hız bozunum bilgilerini kullanarak (açık) bir bit atanması yapmak, birden fazla kodlama yürütümünü veya üretilen bit katarlarının bütün hız-bozunum değerleri için etiketlenip sonradan atama kararlarına göre etiket noktalarından kırılmasını gerektirmektedir. Bu yaklaşımın sebep olduğu gecikme veya sistem karmaşıklığı artışına karşı bütün tel file modellerine ait bütün bölgelerin CSPECK küme bölüntüleme kodlayıcısı ile beraber, ortak kodlandığı bir yaklaşım düşünüyoruz. Burada CSPECK küme bölüntüleme kodlayıcısı sadece bir tel filenin bölgelerine değil ayrıca değişik tel file modellerine de (örtük) bit atanması gerçekleştirebilecektir. Önerilen örtük bit atanması yönteminin yukarıda bahsedilen açık bit atanması yöntemi ile hız-bozunum başarımlı karşılaştırması merak konusudur.

Yararlanılan kaynaklar

- BAYAZIT U., Significance Map Pruning and Other Enhancements to SPIHT Image Coding Algorithm, EURASIP Signal Processing: Image Communication, 18(11), 769-785, (2003).
- BAYAZIT U., Orcay Ö., Konur U., Gürgen F. S., Predictive Vector Quantization of 3-D Polygonal Mesh Geometry by Representation of Vertices in Local Coordinate Systems, J. Vis. Commun. Image R., 18, 341-353 (2007).
- CHOU P. H., Meng T. H., Vertex Data Compression through Vector Quantization, IEEE Trans. on Visualization and Computer Graphics, 8(4), 373-382 (2002).
- DEERING M., Geometry Compression, ACM SIGGRAPH, Los Angeles, (1995) pp:13-20.
- GUMHOLD S., Strasser W., Real Time Compression of Triangle Mesh Connectivity, Proc. SIGGRAPH '98, Orlando, Florida, (1998) pp: 133-140.
- HOPPE H., Progressive Meshes, Proc. SIGGRAPH, (1996) pp: 99-108.
- ISENBURG M., Snoeyink J., Face Fixer: Compressing Polygon Meshes with Properties, Proc. SIGGRAPH '00, New Orleans, Louisiana, (2000), pp: 263-270.
- ISENBURG M., Alliez P., Compressing Polygon Mesh Geometry with Parallelogram Prediction, Visualization, Boston, MA, (2002) pp: 141-146.
- ISLAM A., Pearlman W. A., An Embedded and Efficient Low-Complexity Hierarchical Image Coder, SPIE Visual Communications and Image Processing, San Jose, California, (1999) pp: 294-305.
- KARNI Z., Gotsman C., Spectral Compression of Mesh Geometry, ACM SIGGRAPH, New Orleans, Louisiana, (2000) pp: 279-286.
- KARNI Z., Gotsman C., 3D mesh compression using fixed spectral bases, Graphics Interface, Ottawa, Ontario, Canada, (2001) pp: 1-8.
- KARYPIS G., Kumar V., MeTiS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, Version 4.0, Univ. of Minnesota, Dept. of Computer Science, (1998).
- KHODAKOVSKY A., Schröder P., Sweldens W., Progressive Geometry Compression," ACM SIGGRAPH 2000, New Orleans, Louisiana, pp: 271-278.
- KHODAKOVSKY A., Guskov I., Normal Mesh Compression, Geometric Modelling for Scientific Visualization, Springer-Verlag, Heidelberg, Germany, (2002).
- KONUR U., Bayazıt U., Ateş H. F., Gürgen F. S., Tel File Geometrilerinin Sıradüzensel Küme Bölüntüleme ile Spektral Kodlaması, Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Eskişehir, (2007) pp: 1-4.
- KONUR U., Bayazıt U., Ateş H. F., Gürgen F. S., Spectral coding of mesh geometry with a hierarchical set partitioning algorithm, Visual Communications and Image Processing, San Jose, California, (2008) 682227-68227-8. DOI 10.1117/12.764851.
- LAVU S., Choi H., Baraniuk R.G., Geometry Compression of Normal Meshes using the Estimation Quantization Algorithm, ACM SIGGRAPH/ Eurographics Symposium on Geometry Processing, Aachen, Germany, (2003) pp: 52-61.

- LEE E.-S., Ko H.-S., Vertex data compression for triangular meshes, *Pacific Graphics*, (2000) pp: 225-234.
- PAJAROLA R., ROSSIGNAC J., Compressed Progressive Meshes, *IEEE Trans. Vis. Comp. Graph.* 6 (1), 79-93, (2000).
- PAYAN F., Antonini M., An Efficient Bit Allocation for Compressing Normal Meshes with an Error-driven Quantization, *Geometric Mesh Processing özel bölüm, Computer Aided Geometric Design*, (2005).
- PEARLMAN W. A., Islam A., Nagaraj N., Said A., Efficient, low-complexity image coding with a set-partitioning embedded block coder, *IEEE Trans. Circuits Syst. Video Technology.* 14(11), 1219-1235, (2004).
- ROSSIGNAC J., Edgebreaker. Connectivity compression for triangle meshes, *IEEE Trans. on Visualization and Computer Graphics*, 5, 1, 47-61, (1999).
- SAID A., Pearlman W. A., A New, Fast and Efficient Image Codec based on Set Partitioning in Hierarchical Trees, *IEEE Transactions on Circuits and Systems for Video Technology*, 6, 243-250, (1996).
- SAID A., Introduction to Arithmetic Coding - Theory and Practice, HP Laboratories Technical Report HPL-2004-76, Palo Alto, California, USA, (2004).
- SHAPIRO J.M., Embedded Image Coding using ZeroTrees of Wavelet Coefficients, *IEEE Trans. on Signal Processing*, 41, 3445-3462, (1993).
- SORKINE O., Cohen-Or D., Toledo S., High-pass quantization for mesh encoding, *Eurographics/ACM SIGGRAPH Symp. on Geometry Processing*, Aachen, Germany, (2003) pp: 42-51.
- SZYMCZAK A., Rossignac J. ve King D., Piecewise Regular Meshes: Construction and Compression, *Graphical Models*, (2002).
- TAUBIN G., Rossignac J., Geometric Compression Through Topological Surgery, *ACM Trans. Graphics*, 17(2), 84-115 (1998).
- TAUBIN G., Gueziec A., Horn W., Lazarus F., Progressive Forest Split Compression, *ACM SIGGRAPH*, Orlando, Florida, (1998) pp: 123-132.
- TOUMA C., Gotsman C., Triangle Mesh Compression, *Graphics Interface*, Vancouver, British Columbia, Canada, (1998) pp: 26-34.
- WITTEN I. H., Neal R. M., Cleary J. G., Arithmetic Coding for Data Compression, *Communications of the ACM*, 30(6), pp: 521-540, (1987).

**TÜBİTAK
PROJE ÖZET BİLGİ FORMU**

Proje No: 106E064
Proje Başlığı: Üç Boyutlu Göz Geometrisinin Küme Bölüntüleme ile Verimli Spektral Kodlaması
Proje Yürütücüsü ve Araştırmacılar: Doç. Dr. Uluğ Bayazıt (yürütücü) Hasan Fehmi Ateş (araştırmacı)
Projenin Yürütüldüğü Kuruluş ve Adresi: İşık Üniversitesi, Şile Kampüsü, Kumbaba Mevkii, Şile, İstanbul İstanbul Teknik Üniversitesi, Ayazağa Kampüsü, Ayazağa, İstanbul
Destekleyen Kuruluş(ların) Adı ve Adresi:
Projenin Başlangıç ve Bitiş Tarihleri: 01/09/2006-01/06/2008
Öz (en çok 70 kelime) Literatürde klasik Fourier dönüşümünün üç boyutlu tel file yapılarına genelleştirilmesi ile elde edilmiş olan dönüşümün düzenli olmayan tel file yapılarına uygulanması ile özellikle düşük bit hızlarında yüksek sıkıştırma başarımı sağlanmıştır. Bu projede, ilk aşamada, görüntü kodlamada dalgacık dönüşüm katsayılarının verimli sıkıştırılması için kullanılan sıradüzensel küme bölüntüleme tekniği üç boyutlu tel file yapılarının spektral dönüşüm katsayılarının bütün bit hızlarında verimli sıkıştırılması için uygulanmıştır. Düzensiz tel filenin spektral doğuraylarından oluşturulan bu uyarlamalı dönüşümün hesaplama karmaşıklığı çok yüksek olduğundan literatürde önerilmiş olan düzenli bir tel filenin değişmez spektral taban vektörlerinden oluşturulan dönüşümün düşük hesaplama karmaşıklığı ile geometri kodlamasında kullanılması da incelenmiş ve bu dönüşümün başarımı çok daha yüksek olan yeni bir sürümü de geliştirilmiştir.
Anahtar Kelimeler: Tel file geometri sıkıştırma, tel file geometri kodlama, düzensiz tel file, küme bölüntüleme, spektral sıkıştırma, uyarlamalı dönüşüm, değişmez spektral taban vektörleri, hesaplama karmaşıklığı.
Projeden Yapılan Yayınlar: Umut Konur , Uluğ Bayazıt, Hasan F. Ateş, Fikret S. Gürgen, Spectral coding of mesh geometry with a hierarchical set partitioning algorithm, Visual Communications and Image Processing, San Jose, California (2008) 68227-68227-8. DOI 10.1117/12.764851 . Umut Konur , Uluğ Bayazıt, Hasan F. Ateş, Fikret S. Gürgen, Tel File Geometrilerinin Sıradüzensel Küme Bölüntüleme ile Spektral Kodlaması, Sinyal İşleme ve İletişim Uygulamaları Konferansı, Eskişehir (2007), pp: