

AŞAMALI KAFES KODLU NİCEMLEME İLE İMGE KODLAMA

Ali BİLGİN Philip J. SEMENTILLI[†] Michael W. MARCELLIN

Arizona Üniversitesi Elektrik ve Bilgisayar Mühendisliği Bölümü,

[†]Science Applications International Corp.,

E-posta: {bilgin,mwm}@ece.arizona.edu, sementip@aries.tucson.saic.com

ÖZ

Günümüz imge kodlayıcılarında aranılan önemli özelliklerden biri aşamalı kod çözmeye elverişli olmalarıdır. Kafes Kodlu Nicemleme (KKN), etkili bir nicemleme yöntemi olarak imge kodlama uygulamalarında yaygın olarak kullanılmaktadır. Bu çalışmada KKN'ye aşamalı kod çözme özelliği kazandırmak için yapılması gereken değişiklikler tartışılmakta ve bu özelliğe sahip bir imge kodlayıcı tanıtılmaktadır. Yapılan deneylerde, bu imge kodlayıcının performansının oldukça yüksek olduğu görülmüştür.

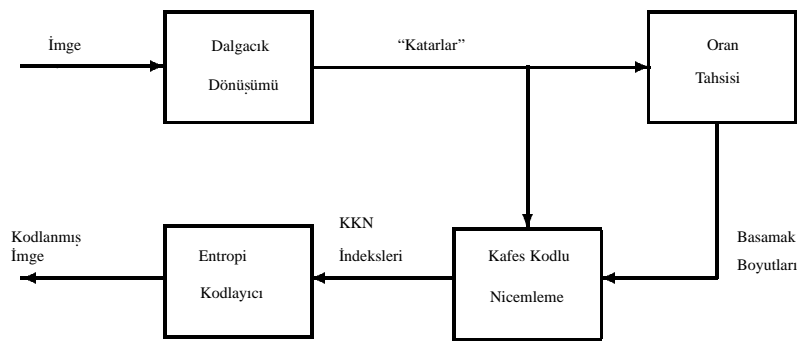
1. SUNUŞ

Günümüzde oldukça yaygın olarak kullanılan ISO imge sıkıştırma standardı JPEG [1], aşamalı kod çözmeye yönelik iki yöntem tanımlamaktadır. Bu yöntemlerden birincisinde, Ayırık Kosinüs Dönüşümü (AKD) katsayılarını, alçak frekanslara ait katsayılardan yüksek frekanslara ait katsayılara doğru kodlayarak, artan spektral çözünürlük elde edilmektedir. İkinci yöntemde ise, AKD katsayıları bit düzlemlerine ayrılarak, en önemli bite karşı gelen düzlemden en önemsiz bite karşı gelen düzleme doğru kodlanmaktadır. Bu yöntemler kodlayıcıya bir çok cazip özellik kazandırır. Örneğin, kodlanmış imgeyi tam olarak çözmeden imgeye göz atmak mümkündür.

Yakın geçmişte, AKD'ne dayalı imge kodlama yöntemleri yerlerini dalgacık dönüşümüne dayalı imge kodlama yöntemlerine bırakmaya başlamıştır. Dalgacık dönüşümü kullanan imge kodlayıcılarına aşamalı iletişim yeteneği kazandırmak için birçok çalışma yapılmıştır [2, 3, 4]. Bu çalışmalarda dalgacık dönüşümü katsayıları önce sayıl olarak nicemlenmiş, daha sonra çeşitli entropi kodlama teknikleri kullanılarak kodlanmıştır. Bu çalışmada ise, dalgacık dönüşümü katsayıları KKN kullanılarak nicemlendiğinde aşamalı iletişim için yapılması gerekenler tartışılmaktadır. Sunulan imge kodlayıcı geliştirilmekte olan JPEG-2000 imge sıkıştırma standardı için önerilmiş ve ilk aşama testlerinde çok başarılı sonuçlar almıştır.

Şekil 1'de dalgacık/KKN tabanlı imge kodlayıcı görülmektedir. Kodlanacak imgeye önce dalgacık dönüşümü uygulanır. Daha sonra altbantlara ait dalgacık katsayıları satır satır taranarak veri katarları oluşturulur. Her bir katarın hangi oranda kodlanması gerektiği hesaplandıktan sonra katarlar tek tek KKN işlemine tabi tutulurlar. Daha sonra, KKN indeksleri uygun bir entropi kodlama yöntemi kullanılarak kodlanır. Bu çalışmada, önce aşamalı KKN incelenmekte, daha sonra KKN indekslerinin aşamalı iletişime elverişli şekilde kodlanması için bir entropi kodlama yöntemi önerilmektedir.

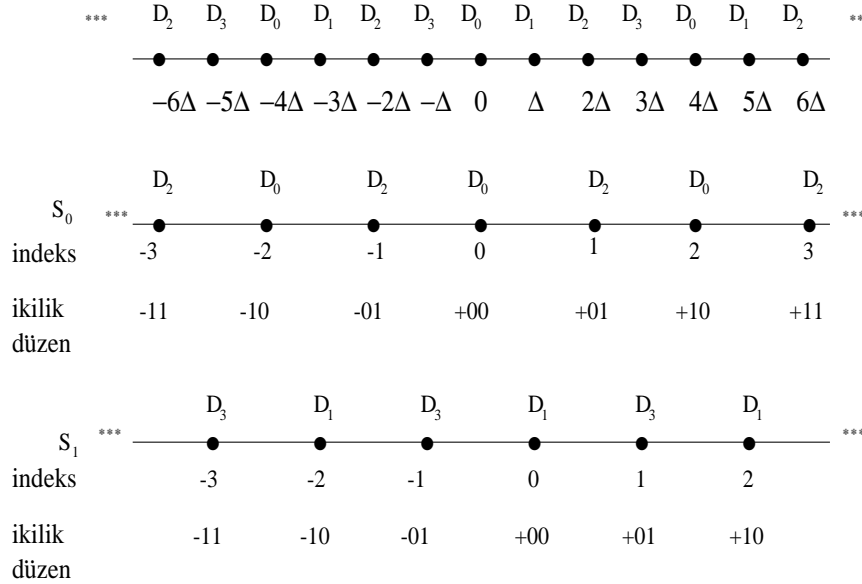
Bu çalışma kısmen A.B.D. Ulusal Bilim Vakfı'nın NCR-9258 374 No.'lu desteği ile gerçekleştirilmiştir.



Şekil 1: İmge Kodlayıcının Blok Şeması.

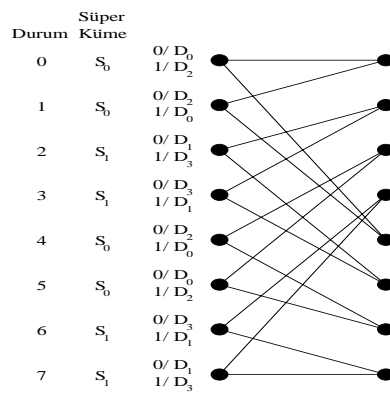
2. AŞAMALI KAFES KODLU NİCEMLEME

KKN genişletilmiş sinyal kümesi ve küme bölme yöntemlerine dayalı etkili ve çok karmaşık olmayan bir nicemleme yöntemidir [5]. İlk aşamada bir sayılı kod kitabı D_0, D_1, D_2 ve D_3 olarak dört alt kümeye ayrılır. Bu işlem Şekil 2’de gösterilmiştir. Bu alt kümeler daha sonra bir kafesin dallarını adlandırmakta kullanılır. Şekil 3’de 8 durumlu bir kafes ve dal adlandırılması gösterilmektedir. Kodlanması gereken bir veri katarının ortalama hatasını azaltmak için kafes üzerinde izlenmesi gereken yol Viterbi algoritması [6] kullanılarak seçilir. Kafes üzerindeki her durumda kod sözcüğü $S_0 = D_0 \cup D_2$ ya da $S_1 = D_1 \cup D_3$ olarak tanımlanan iki süper kümeden birinden seçilir. Örneğin Şekil 3’de, 0 durumunda kod sözcüğü sadece S_0 süper kümesinden seçilebilir. Süper küme içerisinde bir kod sözcüğü seçildiğinde, süper küme indeksi bu sözcüğün hangi alt kümeye ait olduğunu ve dolayısı ile gelecek durumun ne olması gerektiğini belirler. Bundan dolayı, KKN’nin çıkışı süper küme indekslerini gösteren işaretli tamsayılardan oluşmaktadır.



Şekil 2: Kafes Kodlu Nicemleme İndeksleri.

Şekil 2’de de görüldüğü gibi, bir süper kümenin iki alt kümesine ait indeksler ikili düzenlerindeki en önemsiz bitlerinde farklılık gösterir. Başka bir deyişle, en önemsiz bit bir kod sözcüğünün hangi alt kümeye ait olduğunu belirler. Dolayısı ile kod sözcüklerinin en önemsiz bitleri kafes üzerinde izlen-



Şekil 3: 8 Durumlu Kafes.

mesi gereken yolu da belirlemektedir. Kod çözücünün kafes üzerinde izlemesi gereken yol en önemsiz bitler kullanılarak belirlendiği için, bu bitler kod çözücüye ulaşmadan KKN'nin tam olarak tersine çevrilmesi mümkün değildir.

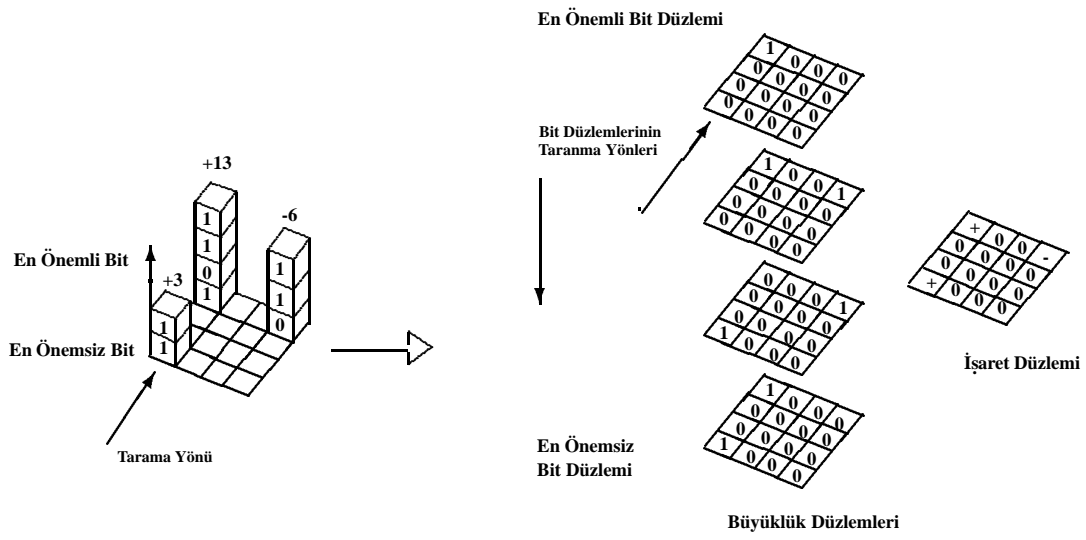
KKN'nin tam olarak tersine çevrilmesi için veri katarının bütün bitlerinin kod çözücüye ulaşmış olması gerektiğinden, KKN'nin ardışıl yaklaşıklama düzenlerine uygun olmadığı izlenimi doğabilir. Ancak, bitlerin sadece bir kısmının kod çözücüye ulaşması durumunda, KKN'nin yaklaşık olarak tersine çevrilmesi mümkündür. O ana kadar kod çözücüye ulaşmış olan bitlerin oluşturduğu işaretli tam sayı, basamak boyunun iki katıyla çarpılırsa ters KKN işlemi yaklaşık olarak gerçekleştirilir. Katar-daki bütün bitler kod çözücüye ulaştığında da, KKN tam olarak tersine çevrilebilir.

3. BİT DÜZLEMİ KODLAYICI

Ardışıl yaklaşıklamada bozunumu daha çok azaltacak bilginin kod çözücüye daha önce gönderilmesi amaçlandığı için, ardışıl yaklaşıklamaya uygun bir kodlayıcı bit düzlemlerini en önemli bit düzleminden en önemsiz bit düzlemine doğru kodlar. Bu çalışmada KKN indeksleri iki değişik metod kullanılarak kodlanmıştır. Bu metodlardan birincisi, [4]'de tanıtılan SPIHT metodudur. İkinci metodda ise, bit düzlemleri, Şekil 4'de görüldüğü gibi, en önemli bit düzleminden, en önemsiz bit düzlemine ve en kaba altbanttan, en ince altbanta doğru taranmakta ve [7]'deki aritmetik kodlayıcı kullanılarak kodlanmaktadır.

Bit düzlemi kodlayıcılarında dalgacık katsayılarının işaretlerinin kodlanma yöntemi de oldukça önem taşır. Kod çözücü bütün katsayıları sıfır olarak varsayarak kod çözmeye başladığı için, bir katsayının işaretine, ancak o katsayının sıfır olmayan en önemli biti kod çözücüye ulaştıktan sonra ihtiyaç duyulmaktadır. Bundan dolayı, işaret bitleri, kodlanan katsayının sıfır olmayan en önemli bitinin hemen arkasından kodlanmaktadır. Bu yöntemde, işaret bitleri kod çözücüye ihtiyaç duyulduğu anda ulaştığı için, yöntem aşamalı kod çözmeye uygundur.

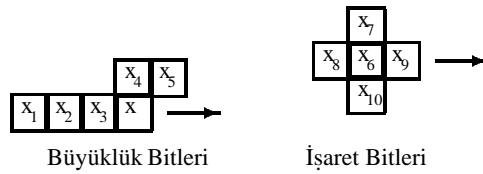
Bit düzlemi kodlayıcısının performansını arttırmak için bir bağlam modeli kullanılmakta ve kodlanacak bitin bağlamı göz önünde bulundurularak aritmetik kodlama olasılık tabloları seçilmektedir. Bağlamın oluşturulmasında hem büyüklük, hem de işaret bitlerinden faydalanılmaktadır. Şekil 5'de bağlam bitleri için kullanılan model gösterilmektedir. Şekilde x kodlanmakta olan dalgacık katsayısının büyüklük, x_6 ise işaret bitini temsil etmektedir. Bağlamın kod çözücüde de oluşturulabilmesi için, bağlam modelinin nedensel olarak tasarlanması gerekmektedir. Bu, Şekil 5'deki büyüklük bitlerinin seçiminde görülmektedir. Bağlam, büyüklük bitlerinden kod çözücüye ulaşmış olanlar arasından seçilen bitler kullanılarak oluşturulmuştur. Bağlamın oluşturulmasında kullanılan işaret bitleri de nedensel



Şekil 4: Bit Düzlemlerinin Kodlanması .

olarak seçilmek zorundadır. Ancak kodlayıcı çok önemli bit düzlemlerinden az önemli bit düzlemlerine doğru ilerledikçe, tarama yönüne göre nedensel olmayan katsayılarla ait işaret bitleri de kod çözücünde bulunabilir. Dolayısıyla işaret bitleri, büyüklük bitlerinden sağlanamayan, tarama yönüne göre geleceğe yönelik bilgi içermektedirler. Bu özellik de Şekil 5’deki bağlam modelinin tasarımında kullanılmıştır. Şekil 5’de görülen 10 bit aşağıdaki gibi nicemlenerek 6’ya indirilmekte ve 64 adet bağlam oluşturulmaktadır. Aşağıdaki işlemlerde | VEYA işlemini temsil etmektedir.

$$\begin{aligned}\hat{x}_1 &= x_1 \\ \hat{x}_2 &= x_2 \\ \hat{x}_3 &= x_3 \\ \hat{x}_4 &= x_4 \mid x_5 \\ \hat{x}_5 &= x_6 \\ \hat{x}_6 &= x_7 \mid x_8 \mid x_9 \mid x_{10}.\end{aligned}$$



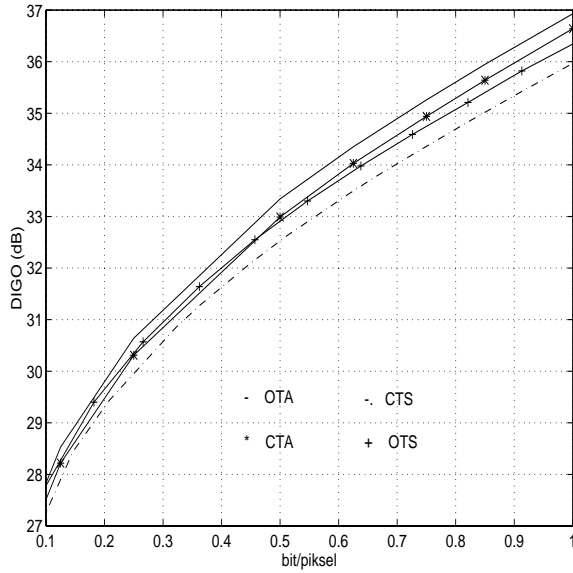
Şekil 5: Bağlam Bitleri

4. KODLAYICI PERFORMANSI

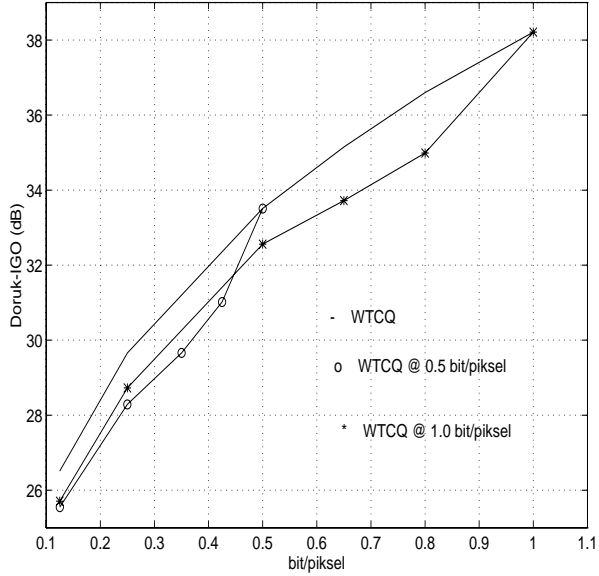
Bu makalede sunulan yöntemler kullanılarak dört değişik sistem oluşturulmuştur: OTA (optimum oran tahsisi/KKN/Bit düzlemi kodlayıcısı), CTA (sabit oran tahsisi/KKN/Bit düzlemi kodlayıcısı), OTS (optimum oran tahsisi/KKN/SPIHT), CTS (sabit oran tahsisi/KKN/SPIHT). Şekil 4’de bu sistemlerin performansı “Goldhill” imgesi kullanılarak karşılaştırılmaktadır. Şekildeki sonuçlar [11]’de tanımlanan

(9,7) süzgeç bankası ve 5 seviyeli ikinci ayırım kullanılarak elde edilmiştir. Yapılan deneylerde OTA düzeninin diğerlerine oranla daha başarılı sonuç aldığı görülmüştür. Buradan sonra sunulan sonuçlarda WTCQ olarak da adlandırılan bu yöntem kullanılmaktadır.

Şekil 4’de WTCQ kodlayıcısının aşamalı iletişim performansı görülmektedir. Şekilde 0.5 ve 1.0 bit/piksel oranlarında kodlanmış bir imgenin aşamalı olarak çözülmesi ile elde edilen Doruk-İşaret-Gürültü-Oranları(Doruk-İGO), değişik oranlarda tamamen çözülmüş imgelerin Doruk-İGO’ları ile karşılaştırılmaktadır. Katarlar arasında optimum oran tahsisi ve yaklaşık ters KKN işlemi yapıldığından, aşamalı olarak çözülen imgeler aynı oranda tam olarak çözülmüş imgelerden daha düşük Doruk-İGO’na sahiptirler.



Şekil 6. Çeşitli Kodlayıcıların Karşılaştırılması .



Şekil 7. WTCQ’nun Aşamalı İletişim Performansı .

Tablo 1’de WTCQ “Barbara” imgesi kullanılarak diğer yüksek performanslı imge kodlama yöntemlerinden SPIHT [4], ve SFQ [8] ile karşılaştırılmaktadır. Tablodaki tüm sonuçlar [9]’da sunulan (10,18) dalgacık süzgeçleri kullanılarak elde edilmiştir. WTCQ sonuçları önce 3 seviye düzgün ayırışım, daha sonra da referans altbandının 2 seviye ikinci ayırışımı ile elde edilmiştir. Diğer iki yönteme ait sonuçlar ise dalgacık paketleri kullanılarak elde edilmiştir [10]. Tablo 2’de de aynı yöntemler “Gold-hill” imgesi, [11]’de sunulan (9,7) dalgacık süzgeçleri ve 5 seviye ikinci dönüşüm kullanılarak karşılaştırılmaktadır.

Tablo 1: “Barbara” İmgesi için Doruk-İşaret-Gürültü-Oranı (DİGO) Sonuçları.

Yöntem	Ayırışım	DİGO (dB)		
		0.25 b/p	0.5 b/p	1.0 b/p
WTCQ	3+2	29.66	33.51	38.21
SFQ	Dalgacık Paketi	29.67	33.51	37.96
SPIHT	Dalgacık Paketi	29.36	33.07	37.71

Sunulan kodlayıcı aynı zamanda kayıpsız kodlamaya da uygundur. Kayıpsız kodlama için, kayıplı olarak kodlanmış bir imge ile orijinal imge arasındaki fark bit düzlemi kodlayıcısı kullanılarak kodlanmakta ve kayıplı kodlanmış imgenin arkasına eklenmektedir. Tablo 3’de WTCQ ile CREW [3] kodlayıcılarının kayıpsız kodlama performansları karşılaştırılmaktadır.

Tablo 2: “Goldhill” İmgesi için Doruk-İşaret-Gürültü-Oranı (DİGO) Sonuçları.

Yöntem	Ayrışım	PSNR (dB)		
		0.25 bpp	0.5 bpp	1.0 bpp
WTCQ	İkici	30.67	33.35	36.94
SFQ [8]	İkici	30.71	33.37	36.70
SPIHT [4]	İkici	30.56	33.13	36.55

Tablo 3: WTCQ ve CREW’ın Kayıpsız Kodlama Performansları .

Yöntem	İmge		
	Lenna	CT	X-Ray
CREW	4.35	4.11	6.06
WTCQ	4.41	4.10	6.16

5. KAYNAKÇA

- [1] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1992.
- [2] J. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients”, *IEEE Transactions on Signal Processing*, cilt 41, sayfa 3445–3462, Aralık 1993.
- [3] E. L. Schwartz, A. Zandi, and M. Boliek, “Implementation of compression with reversible embedded wavelets”, in *Proc. of SPIE, Applications of Digital Image Processing XVIII*, Temmuz 1995, cilt 2564.
- [4] A. Said and W. A. Pearlman, “A new fast and efficient image codec based on set partitioning in hierarchical trees”, *IEEE Transactions on Circuits and Systems for Video Tech.*, cilt 6, sayfa 243–250, Haziran 1996.
- [5] M. W. Marcellin and T. R. Fischer, “Trellis coded quantization of memoryless and Gauss-Markov sources”, *IEEE Transactions on Communications*, cilt 38, sayfa 82–93, Ocak 1990.
- [6] Jr. G. D. Forney, “The Viterbi algorithm”, *Proc. IEEE*, cilt 61, sayfa 268–278, Mart 1973.
- [7] I. H. Witten, R. M. Neal, and J.G. Cleary, “Arithmetic coding for data compression”, *Communications of the ACM*, cilt 30, sayfa 520–540, Haziran 1987.
- [8] Z. Xiong, K. Ramchandran, and M. T. Orchard, “Space-frequency quantization for wavelet image coding”, *IEEE Transactions on Image Processing*, cilt 6, sayfa 677–693, Mayıs 1997.
- [9] M. J. Tsai, J. D. Villasenor, and F. Chen, “Stack-run image coding”, *IEEE Transactions on Circuits and Systems for Video Tech.*, cilt 6, sayfa 519–521, Ekim 1996.
- [10] Z. Xiong, K. Ramchandran, and M. T. Orchard, “Wavelet packet image coding using space-frequency quantization”, *IEEE Transactions on Image Processing*’de yayınlanacak.
- [11] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform”, *IEEE Transactions on Image Processing*, cilt 1, sayfa 205–220, Nisan 1992.