

**T. C.**  
**EGE ÜNİVERSİTESİ**  
**ULUSLARARASI BİLGİSAYAR ENSTİTÜSÜ**

**UBI 614**  
**VIDEO KODLAYICI TASARIMI**

Prof. Dr. E. Turhan TUNALI

**H.264 ÖLÇEKLENEBİLİR VIDEO KODLAYICISI**  
**İÇİN KODLAMA PARAMETRELERİ**  
**VE**  
**BAŞLICA KODLAMA PARAMETRELERİNİN**  
**KODLANMIŞ VIDEO KALİTESİ ÜZERİNDEKİ ETKİLERİ**

**Hazırlayanlar:**  
Sinem ASLAN  
Gül Boztok ALGIN

Haziran 2008



## İçindekiler:

|       |                                                                                           |    |
|-------|-------------------------------------------------------------------------------------------|----|
| 1     | Giriş .....                                                                               | 4  |
| 2     | H.264 Scalable Extension SVC .....                                                        | 4  |
| 2.1   | H.264 SVC Tarihçesi.....                                                                  | 4  |
| 2.2   | H.264 SVC Kodlayıcı Yapısı .....                                                          | 4  |
| 2.3   | Ölçeklenebilirlik Çeşitleri: .....                                                        | 6  |
| 2.3.1 | Zamansal Ölçeklenebilirlik ( <i>Temporal Scalability</i> ) .....                          | 8  |
| 2.3.2 | Uzamsal Ölçeklenebilirlik ( <i>Spatial Scalability</i> ).....                             | 8  |
| 2.3.3 | Kalite Ölçeklenebilirliği ( <i>SNR Scalability</i> ) .....                                | 9  |
| 3     | SVC Kodlama parametreleri .....                                                           | 12 |
| 3.1   | Temel Konfigürasyon Dosyası .....                                                         | 12 |
| 3.1.1 | Genel Parametreler .....                                                                  | 13 |
| 3.1.2 | Kodlama Yapısı Parametreleri .....                                                        | 14 |
| 3.1.3 | Hareket Arama Parametreleri.....                                                          | 15 |
| 3.1.4 | Loop Filtresi Parametreleri.....                                                          | 16 |
| 3.1.5 | Katman Tanım Parametreleri .....                                                          | 16 |
| 3.1.6 | Ağırlıklı Tahminleme Parametreleri .....                                                  | 17 |
| 3.1.7 | Kayıp-Farkındalıklı RDO Parametreleri .....                                               | 17 |
| 3.1.8 | Diğer Parametreler .....                                                                  | 17 |
| 3.2   | Katman Konfigürasyon Dosyası.....                                                         | 18 |
| 3.2.1 | Girdi / Çıktı Parametreleri.....                                                          | 20 |
| 3.2.2 | Kodlama Parametreleri.....                                                                | 21 |
| 3.2.3 | Kontrol Parametreleri.....                                                                | 23 |
| 3.2.4 | Hareket Parametre Dosyası .....                                                           | 24 |
| 3.2.5 | Genişletilmiş Uzamsal Ölçeklenebilirlik Parametreleri .....                               | 24 |
| 3.2.6 | Uyarlanabilir-Referans FGS Parametreleri .....                                            | 26 |
| 3.2.7 | FMO ve Dilimler .....                                                                     | 27 |
| 4     | Başlıca SVC kodlama parametrelerinin kodlama etkinliğine etkisi .....                     | 29 |
| 4.1   | Farklı SNR alternatiflerinin kodlamaya etkileri: .....                                    | 29 |
| 4.2   | Farklı tahminleme yapılarının kodlamaya etkileri:.....                                    | 32 |
| 4.3   | Maksimum yapısal gecikmenin farklı GOP büyüklüklerinde kodlama kalitesine etkileri: ..... | 37 |
| 5     | Sonuç .....                                                                               | 39 |
| 6     | Kaynaklar .....                                                                           | 39 |

# 1 Giriş

Son yıllarda, video akışlı iletim uygulamalarının kullanımının yaygınlaşması ile birlikte, bir tek bit akışının ölçeklenebilir bir şekilde kodlanarak, farklı ağ gereksinim ve uçbirim isteklerine cevap vermesi üzerine gerçekleştirilmekte olan Ölçeklenebilir Video Kodlama (*Scalable Video Coding*) çalışmaları önem kazanmıştır. Bu proje kapsamında, ISO/IEC Moving Pictures Experts Group (MPEG) ve ITU-T Video Coding Experts Group (VCEG)'a ait Ölçeklenebilir Video Kodlama projesinin JSVM 6.8.2 (Ekim.2006 versiyonu) yazılımı kullanılarak, önemli görülen konfigürasyon parametrelerinin PSNR/Bit-hızına etkisi gözlemlenmiştir. Testlerde, iki farklı video akışı için maksimum yapısal gecikme, zamansal ve SNR Ölçeklenebilirlik parametrelerinin video kodlama üzerindeki etkisi araştırılmıştır.

2. bölümde H.264 SVC tarihçesi ve kodlayıcı yapısı hakkında bilgi verildikten sonra, Bölüm 2.3'de ölçeklenebilirlik kavramı ve çeşitleri işlenecektir. Bölüm 3'de, JSVM 6.8.2 yazılımında kullanılan temel ve katman konfigürasyon dosyalarında yer alan tüm parametreler, aldıkları *default* değerler ile birlikte açıklandıktan sonra Bölüm 4'de, önemli görülen parametreler ile gerçekleştirilen test sonuçlarına yer verilecektir.

## 2 H.264 Scalable Extension SVC

H.264 Ölçeklenebilir Video Kodlayıcı (*Scalable Video Coding - SVC*), MPEG 4 Recommendation 10 | ISO / IEC International Standard ISO/IEC 14496-10 AVC olarak da adlandırılan, ITU-T Recommendation H.264 kodlayıcısının ölçeklenebilir versiyonudur. SVC, H.264 AVC' ye uzamsal (*spatial*), zamansal (*temporal*) ve SNR ölçeklenebilirlik getirmektedir ve böylece, çeşitli ölçeklenebilir katmanlar esnek bir şekilde bir araya getirilerek farklı bit iletim hızlarına uyum kolaylaştırılır. Bu sayede, ağ koşullarından veya farklı kapasitelere sahip uçbirimlerden kaynaklanan zorluklara uyum sağlanabildiği gibi, kullanıcıların birbirinden farklı tercihlerine veya ihtiyaçlarına da cevap verilmiş olur.

### 2.1 H.264 SVC Tarihçesi

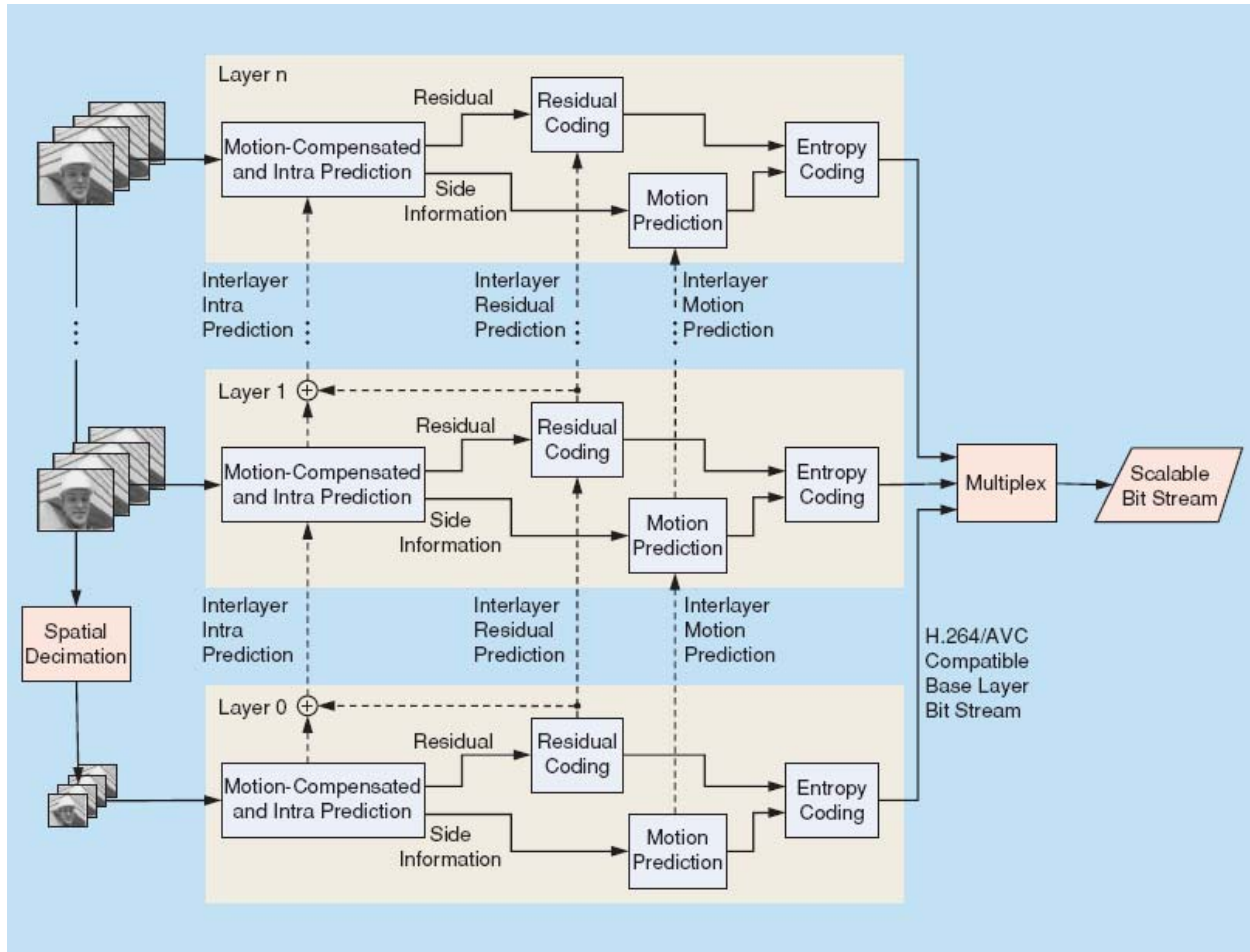
SVC geliştirilmesi yaklaşık 20 yıllık bir tasarı olmakla beraber, MPEG grubunun Aralık 2001'de başladığı çalışmalara kadar, bu konuda tatmin edici çözümler elde edilememiştir. Ekim 2003'te MPEG grubu tarafından ölçeklenebilir video kodlayıcısı teknolojileri için öneri çağrısı yapılmıştır. Ekim 2004' de gelen öneriler arasında yapılan değerlendirme sonucu, HHI bünyesindeki Image Communication Group'una ait öneri geliştirilmeye uygun görülmüştür. Ocak 2005'te, ISO/IEC Moving Picture Experts Group (MPEG) ve ITU-T Video Coding Experts Group (VCEG) gruplarının oluşturduğu ortak çalışma grubu olan JVT takımının çalışması olarak ortaya çıkmıştır [1].

### 2.2 H.264 SVC Kodlayıcı Yapısı

Şekil 1'de,  $n$  uzamsal katmandan oluşan örnek bir kodlayıcı yapısı görülmektedir. Her uzamsal veya kaba-taneli SNR katmanında, hareket dengelemeye dayalı tahminleme (*motion-compensated prediction*) ve intra tahminleme gibi temel kavramlar, H.264 / MPEG4-AVC'de olduğu gibi gerçekleştirilmektedir. Buna ek olarak, farklı katmanlar arasındaki fazlalık bilgisi, hareket parametreleri, intra ve fark verisi için tahminleme mekanizmaları içeren katmanlar-arası (*inter-layer*) tahminleme tarafından kullanılmaktadır. Her katman için girdi resimlerinin temel gösterimi, H.264 / MPEG4-AVC'dakine benzer bir dönüşüm kodlama (*transform coding*) işlemi

yapılarak elde edilir ve en küçük katmana ait NAL birimleri, tek katmanlı H.264 / MPEG4-AVC ile uyumludur. Ayrıca, bu temel gösterimlerin yeniden oluşturulma kalitesi, aşamalı ayrıştırma dilimlerinin (*progressive refinement slices*) kodlanması ile geliştirilebilir ve tüm diğer dilim verisi içeren NAL birimlerinin tersine bu NAL birimleri, ince-taneli kalite ölçeklenebilirliğini (*fine granular quality scalability*) veya esnek bit-hızı uyumunu desteklemek üzere gelişigüzel bir şekilde kesilebilir.

SVC tasarımının önemli bir özelliği, ölçeklenebilirliğin bit-akış seviyesinde sağlanmasıdır. Düşürülmüş uzamsal ve/veya zamansal çözünürlük, global bir SVC bit-akışından, hedeflenen çözünürlükte kod çözümü için gerekli olmayan NAL birimleri (veya ağ paketleri) atılarak basitçe elde edilebilir. Buna ek olarak, aşamalı ayrıştırma dilimlerinin NAL birimleri, bit-hızını ve bununla ilişkili yeniden oluşturma kalitesini daha fazla düşürmek üzere kesilebilir. Kolay bit-akış değişimlerini yönetebilmek için, H.264/MPEG4-AVC'nin 1-byte'lı NAL birim başlıkları, SVC NAL birimleri için 2 byte arttırılmıştır. Bu fazlalık byte lar, belirli bir uzamsal-zamansal çözünürlükte ve kalitede (veya bit hızında) kod çözümü için NAL biriminin gerekli olup olmadığı sinyalini verdiği gibi, NAL biriminin kesilebilir olup olmadığını da bildirir [2].

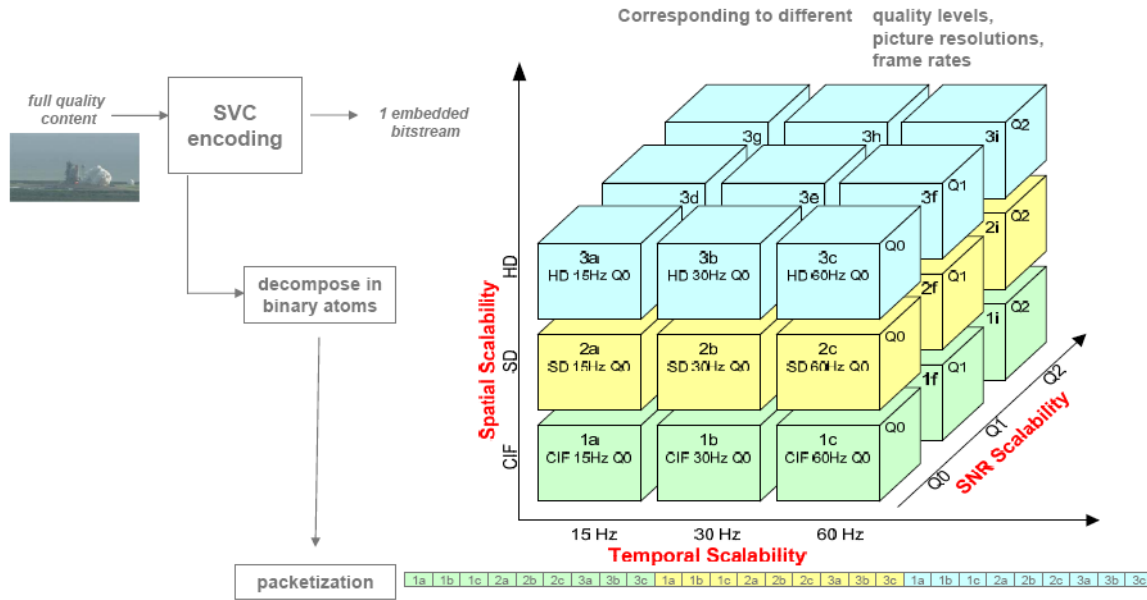


Şekil 1. n uzamsal katman içeren örnek kodlayıcı yapısı [4]

## 2.3 Ölçeklenebilirlik Çeşitleri:

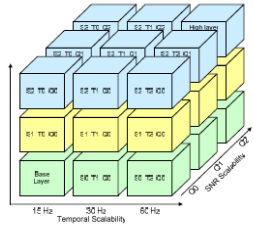
Bir video bit akışının parçaları silindiğinde, sonuç alt-akış yapıları bir takım hedef kod-çözümleri için geçerli bit akışları oluyor ve bu sonuç alt-akış yapıları kaynak içeriğini, kalan verinin miktarı düşünüldüğünde yüksek, fakat orjinal bit akışının tamamının sahip olduğundan daha düşük yeniden oluşturma kalitesinde temsil ediyorsa, bu video bit akışı “ölçeklenebilir,, şeklinde nitelendirilebilmektedir. Bu özelliği taşımayan bit akışları tek-katmanlı bit akışları olarak isimlendirilmektedir.

Ölçeklenebilirlik çeşitleri, uzamsal (*spatial*), zamansal (*temporal*) ve kalite ölçeklenebilirliği şeklinde gruplanır. Uzamsal ölçeklenebilirlik bit akışının altkümelerinin, kaynak içeriğini küçültülmüş resim boyutu (*uzamsal çözünürlük*) ile temsil etmesi durumuna karşılık gelirken, zamansal ölçeklenebilirlik azaltılmış çerçeve hızı (*zamansal çözünürlük*) ile temsil etmesi durumuna karşılık gelmektedir. Kalite ölçeklenebilirliği ile, alt-akış, bit akışının tamamı ile aynı uzamsal-zamansal çözünürlük fakat daha düşük benzerlik (*fidelity*) – ki genellikle signal-to-noise oranı (SNR)’a karşılık gelmektedir – sağlamaktadır. Kalite ölçeklenebilirliği genellikle SNR ölçeklenebilirliği olarak isimlendirilmektedir. Ayrıca farklı ölçeklenebilirlik türleri birleştirilerek, bir tek bit akışı içerisinde, farklı uzamsal-zamansal çözünürlükler ve bit hızları desteklenebilir. Şekil 2’de bir bit akışı için, bir tek kodlayıcı ile gerçekleştirilen ölçeklenebilir kodlama sonrasında, farklı özelliklerdeki kod çözümlerde elde edilen çözülmüş görüntüler Şekil 3’de incelenebilir.

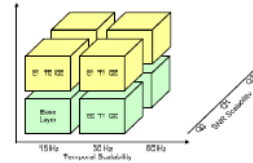


Şekil 2. Esnek birleştirilmiş ölçeklenebilirlik kavramı [5]

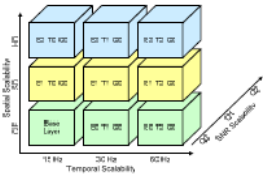
**High resolution**  
**High frame rate**  
**High quality**



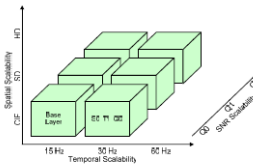
**Medium resolution**  
**Medium frame rate**  
**Medium quality**



**High resolution**  
**High frame rate**  
**Low quality**



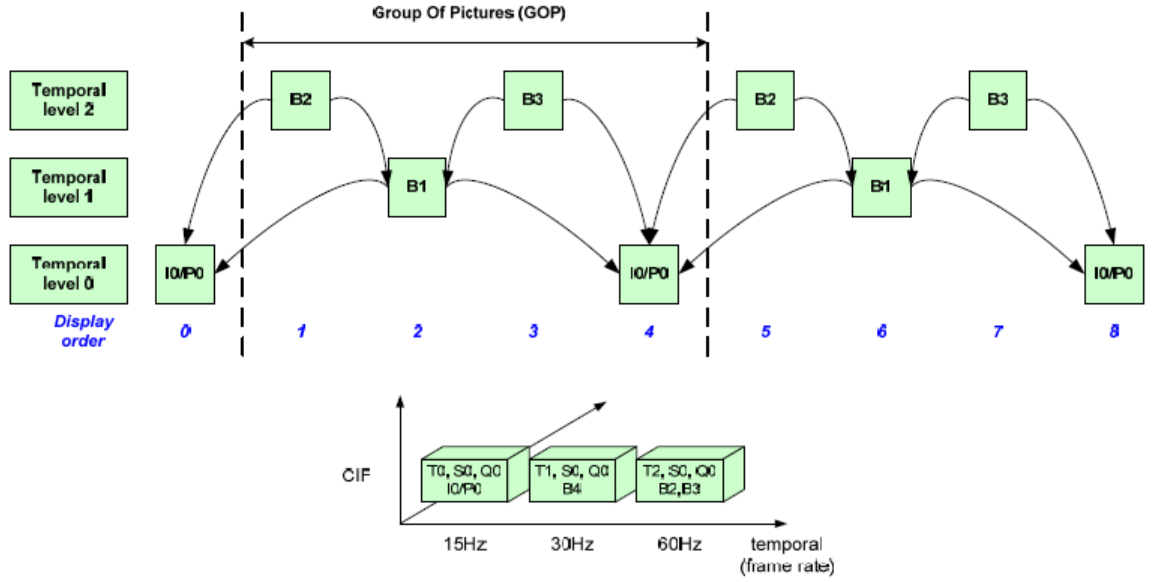
**Low resolution**  
**Medium frame rate**  
**High quality**



**Şekil 3.** Çoklu kod çözme [5]

### 2.3.1 Zamansal Ölçeklenebilirlik (*Temporal Scalability*)

Zamansal ölçeklenebilirlik, bir tek bit akışının, çoklu çerçeve hızlarını desteklemesini sağlayan bir tekniktir. H.264/AVC ve SVC’de hiyerarşik P/B-resimler ve MCTF kullanılarak farklı çerçeve hızlarında kodlama yapılabilmektedir. Hiyerarşik P/B-resimler kullanılarak 3 katmanlı yapıda zamansal ölçeklenebilirliğin elde edilmesine bir örnek Şekil 4’de verilmektedir. Temel katman (*layer 0*) çok az sayıda çerçeve içerir. Üst katmanlar daha fazla çerçeve içerdiğinden üst katmanları çözebilen kod çözücüler daha fazla çerçeve sayısına sahip olacaklarından daha iyi görüntü elde ederler.

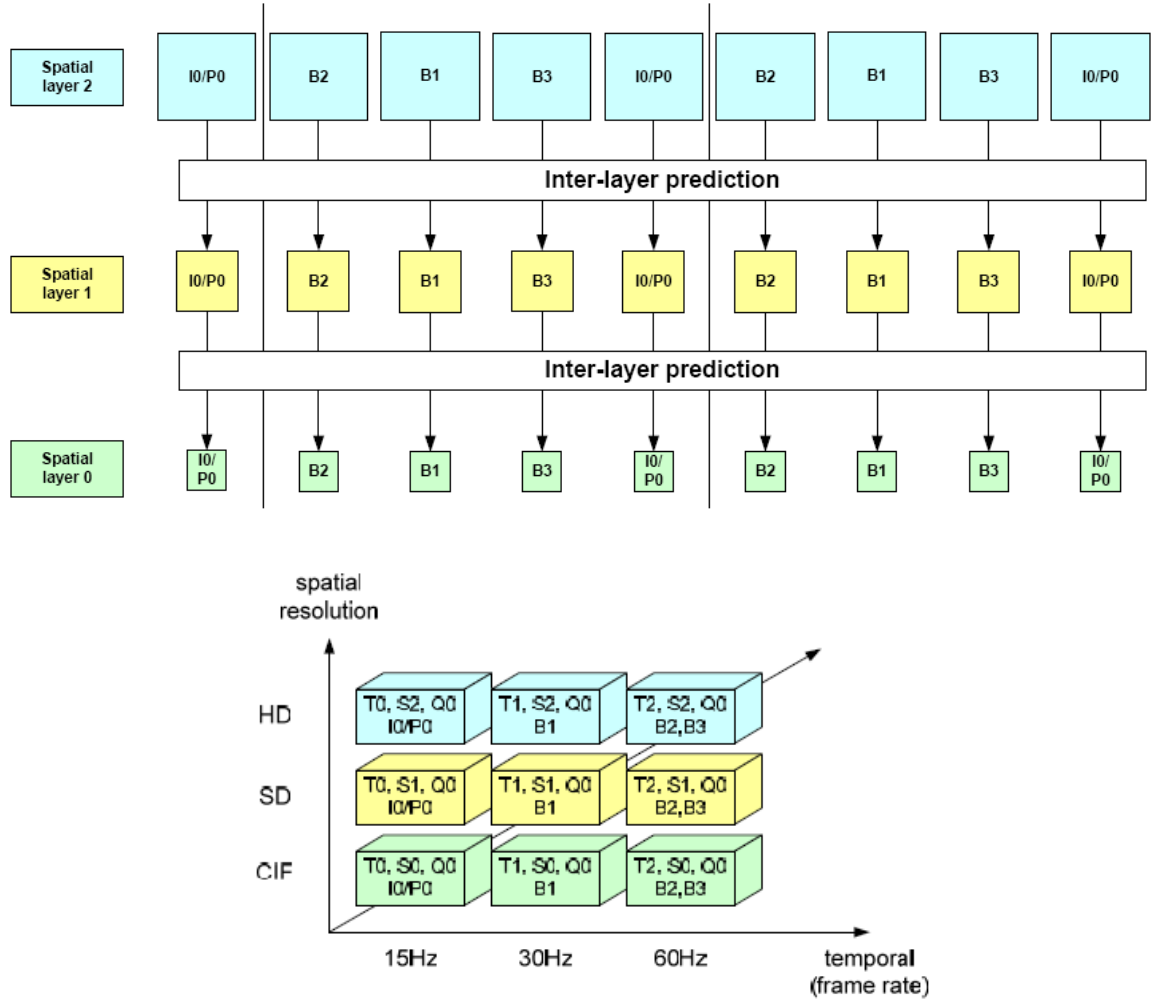


Şekil 4. P/B hiyerarşik resimler ile zamansal ölçeklenebilirlik (H.264 / AVC de desteklemektedir.) [5]

### 2.3.2 Uzamsal Ölçeklenebilirlik (*Spatial Scalability*)

Her uzamsal katman içerisinde, hareket dengelemeye dayalı tahminleme ve intra-tahminleme tek-katmanlı kodlama yapıyormuş gibi gerçekleştirilir. Yani, farklı uzamsal katmanlardaki resimler, katmana özel hareket parametreleri ile bağımsız olarak kodlanır. H.264 AVC nin bu temel kodlama özelliklerine ek olarak, SVC ayrıca katmanlar-arası (*inter-layer*) tahminleme yöntemi kullanmaktadır. Bu yöntem ile, farklı katmanlar arasındaki istatistiksel bağımlılıklar genişletilerek geliştirme katmanlarının kodlama etkinliği (*bit hızı düşürülerek*) geliştirilmektedir. Bu tahminleme mekanizmaları, anahtarlanabilir bir şekilde gerçekleştirilmektedir, böylece kodlayıcı, etkin bir geliştirme katmanı kodlama için hangi temel katmanı kullanacağı seçimini rahatça gerçekleştirir [1-2].





Şekil 5. Uzamsal Ölçeklenebilirlik [5]

### 2.3.3 Kalite Ölçeklenebilirliği (SNR Scalability)

SNR ölçeklenebilirlik, kaba-taneli ve ince-taneli ölçeklenebilirlik şeklinde gerçekleştirilir.



Şekil 6. SNR Ölçeklenebilirlik [5].

### 2.3.3.1 Tek Katmanlı Kodlama (*Single Layer -SL*)

Proje için, CGS ve FGS ile birlikte tek katmanlı kodlama ile elde edilen bit-hızı/PSNR değerlerine de yer verilerek, bu üç çeşit kodlamanın sonucu karşılaştırılmıştır. SL ile sadece tek bir katman üzerinde farklı nicemleme değerleri kullanılarak kodlama yapılmaktadır.

### 2.3.3.2 Kaba-Taneli Ölçeklenebilirlik (*Coarse-grain SNR Scalability – CGS*)

CGS Kalite Ölçeklenebilirliği, uzamsal ölçeklenebilirliğin, temel ve geliştirme katmanı için aynı resim boyutları kullanan özel bir durumu gibi düşünülebilir. CGS'nin farkı, katmanlar-arası tahminlemedeki yukarı örnekleme (*upsampling*) ve ölçekleme işlemlerinin gerçekleştirilmemesidir. Katmanlar-arası tahminleme yapılırken, geliştirme katmanındaki fark resmi verisi yeniden nicemlenirken, önceki CGS katmanı için kullanılan nicemleme parametresinden daha küçük değerdeki nicemleme parametresi kullanılır. CGS kodlama için kullanılan yeniden nicemleme kavramı, birbirini takip eden kalite katmanları arasındaki bit-hız oranı küçük olduğunda (örneğin bir katmandan diğerine %25 artış olduğunda) etkinliğini kaybetmektedir [2].

### 2.3.3.3 İnce-Taneli Ölçeklenebilirlik (*Fine-grain SNR Scalability – FGS*)

İnce-taneli ölçeklenebilirlik ile, kodlama sırasında veya sonrasında geliştirme katmanı kesilerek veya kırılarak (bit hızı ve kodçözümü kalitesi azaltılarak) iletim bit hızı üzerinde yüksek esneklikte kontrol elde edilmektedir ve bu ölçeklenebilirlik, kullanılabilir iletim bant genişliğinin yeterli ölçüde bilinemediği durumlarda oldukça kullanışlı olmaktadır [6].

“Tanecikli (granularity),, terimi, bit hızının hangi duyarlılıkla kontrol edilebileceğini göstermektedir ve dolayısıyla, “İnce-taneli ölçeklenebilirlik (*Fine-grain Scalability*),, bit hızının küçük artışlar şeklinde eklenmesine olanak tanımaktadır [7].

FGS ile, CGS nin dezavantajlarını ortadan kaldırmak ve ince-taneli SNR ölçeklenebilirliği desteklemek için aşamalı ayrıştırma dilimleri (*progressive refinement slices*) kavramı önerilmiştir. Her aşamalı ayrıştırma dilimi, fark verisinin, nicemleme adım büyüklüğünün ikiye bölünmüş haline karşılık gelen (6 artışı QP) bir ayrışımını temsil etmektedir. Bu veri, kod çözücü tarafında, her dönüşüm bloğu için sadece bir ters dönüşüm gerçekleştirilecek şekilde gösterilir. Dönüşüm katsayıları seviyelerinin, aşamalı ayrıştırma dilimleri içerisindeki sıralanışı, bunlara karşılık gelen NAL birimlerinin herhangi gelişigüzel bir noktada kesilmesine olanak tanır ve böylece SNR temel katmanının kalitesi ince-taneli bir şekilde ayrıştırılabilir. Burada, her aşamalı ayrıştırma diliminin, temel resmin sadece fark verisini ayrıştırdığına dikkat edilmelidir.

### 2.3.3.4 Orta-Taneli Ölçeklenebilirlik (*Medium-grain SNR Scalability – MGS*)

SVC tasarımında, özellikle bit akış ve bozulum uyumunun esnekliğini arttırırken, bir takım bit hızlarını sağlaması gereken bit akışlarının kodlama etkinliğini geliştirmek için, bir CGS yaklaşımı varyasyonu olan orta-taneli kalite ölçeklenebilirliği (medium-grain quality scalability)

kavramına yer verilmiştir. CGS kavramından farkı, herhangi bir erişim birimi içerisindeki farklı MGS katmanları arasında anahtarlama imkanı sağlayan modifiye edilmiş bir yüksek-seviyeli sinyalleşme, ve hiyerarşik tahminleme yapıları için, sürüklenme (drift) ve geliştirme katmanı kodlama etkinliği arasında uygun bir ödünleşim ayarlamasına izin veren modifiye edilmiş anahtar resim kavramını içermesidir [3]. Projede gerçekleştirilen testler, MGS içermeyen JSVM 6.8.2 (Ekim, 2006) versiyonunda gerçekleştirilmiştir. MGS, Nisan.2008 tarihinde çıkartılmış olan JSVM 9.12.2 versiyonunda yer almaktadır.

## 3 SVC Kodlama parametreleri

### 3.1 Temel Konfigürasyon Dosyası

```
# JSVM Main Configuration File

#===== 1.GENEL =====
OutputFile          test.264    # Bitstream file
FrameRate           30.0        # Maximum frame rate [Hz]
MaxDelay            1200.0       # Maximum structural delay [ms]
FramesToBeEncoded    100        # Number of frames (at input frame rate)
NonRequiredEnable    0          # NonRequiredSEI enable (0:disable, 1:enable)
CgsSNRRRefinement....0          # SNR refinement as 1: CGS (EI, EP, EB) 0: FGS (PR)

#===== 2. KODLAMA YAPISI =====
GOPSize             2           # GOP Size (at maximum frame rate)
IntraPeriod          4           # Intra Period
NumberReferenceFrames 1         # Number of reference pictures
BaseLayerMode        1           # Base layer mode (0 : AVC with larger DPB,
                                # 1:AVC compatible, 2:AVC w subseq SEI)

#===== 3. HAREKET ARAMA =====
SearchMode           4           # Search mode (0:BlockSearch, 4:FastSearch)
SearchFuncFullPel     3           # Search function full pel
                                # (0:SAD, 1:SSE, 2:HADAMARD, 3:SAD-YUV)
SearchFuncSubPel      2           # Search function sub pel
                                # (0:SAD, 1:SSE, 2:HADAMARD)
SearchRange           32         # Search range (Full Pel)
BiPredIter            4           # Max iterations for bi-pred search
IterSearchRange       8           # Search range for iterations (0: normal)

#===== 4. LOOP FİLTRESİ =====
LoopFilterDisable     0           # Loop filter idc (0: on, 1: off, 2:
                                # on except for slice boundaries)
LoopFilterAlphaC0Offset 0         # AlphaOffset(-6..+6): valid range
LoopFilterBetaOffset  0           # BetaOffset (-6..+6): valid range

#===== 5. KATMAN TANIMI =====
NumLayers             2           # Number of layers
LayerCfg              layer0.cfg  # Layer configuration file
LayerCfg              layer1.cfg  # Layer configuration file

#===== 6. AĞIRLIKLIL TAHMİNLEME =====
WeightedPrediction     0           # Weighting IP Slice (0:disable, 1:enable)
WeightedBiprediction    0           # Weighting B Slice (0:disable, 1:explicit,
                                # 2:implicit)

#===== 7. KAYIP-FARKINDALIKLI RDO =====
LARDO                 0           # Loss-aware RDO (0:disable, 1:enable)

#===== 8. DİĞER PARAMETRELER =====
SuffixUnitEnable       1           # Add suffix unit (0: off, 1: on) shall always be on
                                # in SVC contexts (i.e. when there are
                                # FGS/CGS/spatial enhancement layers)
MMCOBaseEnable         1           # MMCO for base representation (0: off, 1: on)
NestingSEI             1           # Nesting SEI message(1:enable, 0:disable )
SceneInfo              1           # scene info SEI message(1:enable, 0:disable )
```

**Tablo 1.** Örnek JSVM 6.8.2 Temel Konfigürasyon Dosyası

JSVM 6.8.2'deki temel konfigürasyon dosyasındaki tüm parametreler Tablo 1'de gösterilmiştir. Bu parametreler ile ilgili ayrıntı bilgi bundan sonraki bölümde verilecektir.

### 3.1.1 Genel Parametreler

#### 3.1.1.1 OutputFile

*String, default: test.264*

Kodlanacak bit-akışı için dosya adı belirtir.

#### 3.1.1.2 FrameRate

*Double, default: 60.0*

Tüm girdi akışı için maksimum çerçeve hızını belirtir. *FramesToBeEncoded* parametresi, *FrameRate* girdi çerçeve hızında kodlanması gereken çerçeve sayısını belirtir. Buna ek olarak, *FrameRate* parametresi, bir katmanın gerçek kodlama yapısını belirlemek için gereklidir. Bir katmanın temel kodlama yapısı yani GOP büyüklüğü, temel konfigürasyon dosyalarında yer alan *FrameRate* ve *GOPSize* parametreleri ve katman konfigürasyon dosyalarında yer alan *FrameRateIn* ve *FrameRateOut* parametreleri ile belirlenir.

#### 3.1.1.3 MaxDelay

*Double, default: 1200.0*

Bu parametre, kodlama işlemi için kabul edilebilir maksimum yapısal gecikmeyi milisaniye (ms) olarak belirtir. Bu maksimum gecikmenin aşılması için, tüm veya birkaç resim için hareket dengelemeye dayalı tahminlemenin (*motion-compensated prediction*) geçmiş referans resimlerine kısıtlanması şeklinde kodlama yapısı uyarlanabilir. Genellikle, kodlama etkinliği, maksimum yapısal gecikme zorlanarak düşürülür.

#### 3.1.1.4 FramesToBeEncoded

*Unsigned Int, default: 1*

Girdi akışının kodlanacak olan toplam çerçeve sayısını belirtir. Kodlanacak olan çerçeve sayısı, *FrameRate* ile verilen çerçeve hızı ile belirlenmektedir. Bu yüzden, katman konfigürasyon dosyalarında yer alan *FrameRateOut* değerine bağlı olarak, gerçekte kodlanmış çerçeve sayısı daha küçük olabilir.

#### 3.1.1.5 NonRequiredEnable

*Flag (0 veya 1), default: 0*

Üretilen bit-akışının gerekli-olmayan resim SEI mesajlarını içerip içermediğini belirtir. Eğer, *NonRequiredEnable* 1'e eşitse, bit-akışında gerekli-olmayan resim SEI mesajları yer almaktadır; aksi taktirde yer almamaktadır.

#### 3.1.1.6 CgsSnrRefinement

*Flag (0 veya 1), default: 0*

SNR geliştirmelerin, EI, EP veya EB (flag = 1 iken) dilimleri veya PR (flag = 0 iken) dilimlerinden hangisinin kullanılarak kodlandığı hakkında bilgi verir. JSVM 9.12.2

(Nisan.2008) versiyonunda, MGS (flag = 1 iken) veya CGS (flag = 0 iken)'den hangisinin kodlandığını belirtmektedir.

### 3.1.2 Kodlama Yapısı Parametreleri

#### 3.1.2.1 GOPSize

*Unsigned Int, default: 1*

Video akışını kodlamak için kullanılacak olan GOP büyüklüğünü belirtir. Bir GOP (*group of pictures*), genellikle P-resmi olarak kodlanmış olan bir anahtar resim ve anahtar resimler arasında yer alan hiyerarşik olarak kodlanmış olan birkaç B-resminden oluşur. *GOPSize* parametresinin 2'nin bir kuvvetine eşit olması gerekmektedir ve *FrameRate* tarafından verilen çerçeve hızında belirlenir. Bu yüzden, katman konfigürasyon dosyasında yer alan *FrameRateOut* parametre değerine bağlı olarak, bir katman için gerçek GOP büyüklüğü daha küçük olabilir. Örneğin, *FrameRate* 30'a, *GOPSize* 16'ya ve *FrameRateOut* 7.5'a eşitse, ilgili katmanı kodlamak için kullanılacak olan gerçek GOP büyüklüğü 4'e eşit olur. Tüm katmanların GOP büyüklüğü, her katman için anahtar çerçeveler zamansal olarak dizilmiş şekilde seçilir. Bu nedenle, katman konfigürasyon dosyalarındaki *FrameRateOut* parametrelerine bağlı olarak, kabul edilebilir *GOPSize* sınırlandırılabilir. Bunun için kabul edilebilir maksimum değer 64'dür.

#### 3.1.2.2 IntraPeriod

*Unsigned Int, default: 2^32-1 (-1'e eşittir)*

Kodlanacak video akışı için intra periyodu belirtir. *IntraPeriod* -1 ( $2^{32}-1$ )'e eşit iken, sadece başlangıç resmi intra kodlanır. *IntraPeriod*'un -1'e eşit olmadığı diğer durumlarda, her *IntraPeriod*-uncu resim (*FrameRate* çerçeve hızında) intra kodlanır. *IntraPeriod* parametresi -1'e veya *GOPSize*'in bir katına eşit olmalıdır.

#### 3.1.2.3 NumberReferenceFrames

*Unsigned Int, default: 1*

List 0 ve 1' de yer alan referans resimleri için kullanılacak maksimum aktif girdi sayısını belirtir. Buna ek olarak, belirli bir çerçeveyi kodlamak için kullanılan aktif girdilerin gerçek sayısı, bir çerçevenin GOP içerisindeki konumuna bağlıdır.

#### 3.1.2.4 BaseLayerMode

*Unsigned Int, default: 3*

Temel katmanın (*base layer*) AVC uyumluluğunu belirtir. Aşağıdaki değerleri alabilmektedir:

0 – Daha geniş DPB büyüklüğünde AVC uyumlu temel katman,

1 – AVC uyumlu temel katman,

2 – Zamansal ölçeklenebilirliği desteklemek için alt-sıra SEI mesajları ile AVC uyumlu temel katman

Bu parametre, varsayılan değer geçerli olmasa bile her zaman mevcut olmalıdır.

### 3.1.3 Hareket Arama Parametreleri

#### 3.1.3.1 SearchMode,

*Int, default: 0*

Uygulanacak olan hareket arama algoritmasını belirtir. *SearchMode* 0'a eşit iken, tam kapsamlı blok arama (*exhaustive search, full search*) gerçekleştirilir. *SearchMode* 4' e eşit iken, bir hızlı hareket arama algoritması gerçekleştirilir. Bu parametre için 0 ve 4 dışındaki değerler tanımlı değildir. Hızlı arama algoritması, kodlama süresini önemli ölçüde azaltarak, karşılaştırılabilir ölçüde hız-bozulma etkinliği sağladığı için tercih edilmektedir.

#### 3.1.3.2 SearchFuncFullPel,

*Int, default: 0*

Tamsayı olan örnek konumlarında hareket arama işlemi sırasında hesaplanan bozulma ölçüsünü belirtir. Aşağıdaki değerler desteklenmektedir:

0 – Luminans bileşeni için farkların mutlak değerinin toplamı (*Sum of absolute differences - SAD*)

1 – Luminans bileşeni için farkların karelerinin toplamı (*Sum of squared differences - SSE*)

2 – Luminans bileşeni için, Hadamard dönüşüm uzayındaki farkların mutlak değerlerinin toplamı

3 – Tüm renk bileşenleri için farkların mutlak değerinin toplamı (*Sum of absolute differences - SAD*)

#### 3.1.3.3 SearchFuncSubPel,

*Int, default: 0*

Alt-örnek konumlarında hareket arama işlemi sırasında hesaplanan bozulma ölçüsünü belirtir. Aşağıdaki değerler desteklenmektedir:

0 – Luminans bileşeni için farkların mutlak değerinin toplamı (*Sum of absolute differences - SAD*)

1 – Luminans bileşeni için farkların karelerinin toplamı (*Sum of squared differences - SSE*)

2 – Luminans bileşeni için, Hadamard dönüşüm uzayındaki farkların mutlak değerlerinin toplamı

#### 3.1.3.4 SearchRange,

*Unsigned Int, default: 96*

Hareket arama için, maksimum arama sınırlarını belirtir. Hızlı arama algoritması seçildiğinde, gerçek arama sınırları daha küçük olabilir.

#### 3.1.3.5 BiPredIter,

*Unsigned Int, default: 4*

Çift-tahminli (*bi-predictive*) bloklar için hareket aramada kullanılacak yineleme sayısını belirtir. *BiPredIter* parametresinin değeri, 2'ye eşit veya 2'den büyük tutulduğunda B-resimler için kodlama etkinliği genellikle artmaktadır.

### 3.1.3.6 IterSearchRange,

*Unsigned Int, default: 8*

Çift-tahminli (*bi-predictive*) bloklar için hareket arama yinelemelerinin arama sınırlarını belirtir. *SearchRange* ile belirtilen arama sınırları içerisinde bir başlangıç araması gerçekleştirildiği için, kodlama etkinliği düşürülmeden bu parametreye oldukça küçük değerler verilebilir.

## 3.1.4 Loop Filtresi Parametreleri

### 3.1.4.1 LoopFilterDisable

*Unsigned Int, default: 0*

*Deblocking* filtresinin nasıl uygulanacağını belirtir. Aşağıdaki değerler desteklenmektedir:

- 0 – *Deblocking* filtresi tüm blok kenarlarına uygulanır
- 1 – *Deblocking* filtresi uygulanmaz.
- 2 – Slice sınırları haricinde tüm blok kenarlarına *deblocking* filtresi uygulanır.
- 3 – İki aşamalı *deblocking* işlemi (ikinci aşamada, slice sınırları filtrelenir).
- 4 – *Deblocking* filtresi tüm luma blok kenarlarına uygulanır, chroma filtrelenmez.
- 5 – *Deblocking* filtresi, slice sınırları haricinde tüm luma blok kenarlarına uygulanır, chroma filtrelenmez
- 6 – Luma için iki aşamalı *deblocking* işlemi gerçekleştirilir (ikinci aşamada, slice sınırları filtrelenir), chroma filtrelenmez.

### 3.1.4.2 LoopFilterAlphaC0Offset

*Int, default: 0*

Loop filtresi için alfa CO ofsetini belirtir.

### 3.1.4.3 LoopFilterBetaOffset

*Int, default: 0*

Loop filtresi için beta ofsetini belirtir.

## 3.1.5 Katman Tanım Parametreleri

### 3.1.5.1 NumLayers

*Unsigned Int, default: 1*

Katman sayısını belirtir. Bir katman, uzamsal (*spatial*) bir katmanı veya kaba-taneli SNR ölçekli (*coarse-grain SNR scalable - CGS*) bir katmanı göstermektedir. Burada, zamansal katmanların *GOPSize* parametresi ile ve kaba-taneli SNR ölçekli katmanların, katman konfigürasyon dosyasında yer alan *NumFGSLayers* parametresi ile belirtildiğinin vurgulanması yararlı olabilir (bir “katman” birkaç FGS katmanı içerebilir). (JSVM 9.12.2 (Nisan.2008) versiyonunda katman konfigürasyon dosyalarında yer alan *MgsVectorMode* parametresi ile orta-taneli ölçekli katmanlar (*medium-grain SNR scalable layers - MGS*) belirtilir.) *NumLayers* parametresi 1- 8 aralığında değer alacaktır. Her katman için, *LayerCfg* parametresi kullanılarak bir katman konfigürasyon dosyası belirtilmelidir.



### 3.1.5.2 LayerCfg

*String, no default value*

Bir katman konfigürasyon dosyasının dosya adını belirtir. Temel konfigürasyon dosyası, *LayerCfg* parametresini, her biri belirli bir katman için katman konfigürasyon dosyasını belirtecek şekilde tam olarak *NumLayers* sayıda içermelidir. Buradaki ilk *LayerCfg* parametresi temel katman (LayerId 0) için katman konfigürasyon dosyası belirtirken, bir sonraki *LayerCfg* parametresi, bir sonraki katman (LayerId 1) için katman konfigürasyon dosyası belirtir, ...vb.

### 3.1.6 Ağırlıklı Tahminleme Parametreleri

#### 3.1.6.1 WeightedPrediction

*Flag (0 or 1), default: 0*

P-resimler için ağırlıklı tahminleme kullanılıp kullanılmadığını belirtir. Bu parametre 0'a eşit olduğunda, P resimler için ağırlıklı tahminleme etkisiz hale getirilir. Bu parametre 1'a eşit olduğunda, P resimler için ağırlıklı tahminleme kullanılır duruma getirilir ve tahminleme ağırlıkları kodlama sırasında hesaplanır.

#### 3.1.6.2 WeightedBiPrediction

*Unsigned Int, default: 0*

B resimler için, ağırlıklı tahminlemenin kullanılıp kullanılmadığını ve kullanılıyorsa ne şekilde kullanıldığını belirtir. Bu parametre 0'a eşit iken, B resimler için ağırlıklı tahminleme etkisiz hale getirilir. Bu parametre 1'e eşit iken, B resimler için ağırlıklı tahminleme kullanılır duruma getirilir ve açık ağırlıklandırma modunda (*explicit weighting mode*) gerçekleştirilir; yani gerekli tahminleme ağırlıkları, kodlama sırasında hesaplanır. Bu parametre 2'ye eşit iken, B resimler için ağırlıklı tahminleme kullanılır duruma getirilir ve örtülü ağırlıklandırma modunda (*implicit weighting mode*) gerçekleştirilir. Örtülü modda, tahminleme ağırlıkları hesaplanmaz ve iletilmez, referans çerçeve ile o an kodlanmakta olan çerçeve arasındaki uzaklıktan (Resim Sıra Sayısı - *Picture Order Count*-yoluyla ölçülür) çıkarılır.

### 3.1.7 Kayıp-Farkındalıklı RDO Parametreleri

#### 3.1.7.1 LARDO

*Flag (0 or 1), default: 0*

Kayıp-farkındalıklı hız-bozulma eniyileştirilmiş makroblok mod kararının (*loss-aware rate-distortion optimized macroblock mode decision*) kullanılıp kullanılmadığını belirtir. Bu parametre, katman konfigürasyon dosyalarındaki ClosedLoop parametresi 0'a eşit değilken değerlendirilir.

### 3.1.8 Diğer Parametreler

#### 3.1.8.1 SuffixUnitEnable

*Bool, default: 1*

Her AVC kodlanmış dilim (slice) NAL biriminden sonra bir sonek birimi eklenip eklenmeyeceğini belirtir. Bu parametre 0'e eşit iken, bit akışına hiç bir NAL birim soneki eklenmezken, bu parametre 1'e eşit iken bir NAL birimi soneki eklenir.

### 3.1.8.2 MMCOBaseEnable

*Bool, default: 1*

Anahtar resimlerin temel temsillerinin yönetimi için MMCO kullanılıp kullanılmayacağını belirtir. Bu parametre 0'a eşit iken, bit akışında temel temsili için MMCO kullanılmazken, bu parametre 1'e eşit olduğunda MMCO'ya yer verilir.

### 3.1.8.3 NestingSEI

*Bool, default: 0*

Bit akışında, ölçeklenebilir içiçe geçmiş SEI mesajlarına yer verilir verilmeyeceğini belirtir. Bu parametre 0'a eşit iken, ölçeklenebilir içiçe geçmiş SEI mesajlarına yer verilmezken, 1'e eşit olduğunda, ölçeklenebilir içiçe geçmiş SEI mesajları içeren AVS SEI mesajlarına yer verilir.

### 3.1.8.4 SceneInfo

*Bool, default: 0*

Bit akışında sahne (scene) bilgisi SEI mesajlarına yer verilir verilmeyeceğini belirtir. Bu parametre 0'a eşit iken, sahne bilgisi SEI mesajlarına yer verilmezken, 1'e eşit olduğunda, NestingSEI 1'e eşit olmalıdır ve bit akışında AVC sahne bilgisi SEI mesajları içeren ölçeklenebilir içiçe geçmiş SEI mesajlarına yer verilir.

## 3.2 Katman Konfigürasyon Dosyası

```
# JSVM Layer Configuration File

#===== GİRİŞ / ÇIKTI =====
SourceWidth      176      # Input  frame width
SourceHeight     144      # Input  frame height
FrameRateIn      15       # Input  frame rate [Hz]
FrameRateOut     15       # Output frame rate [Hz]
InputFile        orig.yuv # Input  file
ReconFile        rec_layer.yuv # Reconstructed file
SymbolMode       0        # 0=CAVLC, 1=CABAC
IDRPeriod        0        # IDR period (should be (GOP size*N))
                  # UseAdaptiveGOP should be 0

#===== KODLAMA =====
ClosedLoop       2        # Closed-loop control (0,1:at highest
                        # rate, 2: at lowest and highest rate)
FRExt            1        # 8x8 trafo (0:off, 1:on)
MaxDeltaQP       1        # Max. absolute delta QP
QP               35.0     # Quantization parameters
NumFGSLayers     0.0      # Number of FGS layers ( 1 layer -> DQP = 6 )
FGSMotion        0        # motion ref in FGS slices (0:off, 1:on)
BaseLayerId      1        # Layerd ID for the base layer
ForceReOrdering  1        # Force RPLR commands (0:off, 1:on)
EncSIPFile       layer1.dat # SIP decision file
```



|                 |             |                                                                                                                                                                        |
|-----------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |             | # 3: Box-out, 4: Raster Scan, 5: Wipe,<br># 6:Explicit,slice_group_id read from<br># SliceGroupConfigFileName                                                          |
| SlcGrpChgDrFlag | 1           | # slice_group_change_direction_flag,<br># 0: box-out clockwise, raster scan or<br># wipe right<br># 1: box-out counter clockwise,reverse raster<br># scan or wipe left |
| SlcGrpChgTrMns1 | 1           | # slice_group_change_rate_minus1                                                                                                                                       |
| SlcGrpCfgFileNm | sg.cfg      | # SliceGroupConfigFileName,<br># Used for slice_group_map_type 0, 2, 6                                                                                                 |
| UseRedundantSlc | 1           | # UseRedundantSlice,<br># 0: not used,<br># 1: one redundant slice used for each slice                                                                                 |
| PLR             | 3           | # (Packet Loss Rate)                                                                                                                                                   |
| NumROI          | 1           | # ????????                                                                                                                                                             |
| ROICfgFileNm    | roiconf.cfg | # ????????                                                                                                                                                             |

**Tablo 2.** Örnek JSVM 6.8.2 katman konfigürasyon dosyası

JSVM 6.8.2'deki katman konfigürasyon dosyasındaki tüm parametreler Tablo 2'de gösterilmiştir. Bu parametreler ile ilgili ayrıntı bilgi bundan sonraki bölümde verilecektir.

### 3.2.1 Girdi / Çıktı Parametreleri

#### 3.2.1.1 SourceWidth

*Unsigned Int, default: 176*

Luma örnekleri şeklindeki girdi resimlerinin genişliğini belirtir. *SourceWidth*'in sıfırdan farklı ve 16'nın katı olması gerekmektedir.

#### 3.2.1.2 SourceHeight

*Unsigned Int, default: 352*

Luma örnekleri şeklindeki girdi resimlerinin uzunluğunu belirtir. *SourceHeight*'in sıfırdan farklı ve 16'nın katı olması gerekmektedir.

#### 3.2.1.3 FrameRateIn

*Double, default: 30*

Girdi akışının çerçeve hızını belirtir. Temel konfigürasyon dosyasındaki *FrameRate* parametresinin *FrameRateIn*' in bir katı ve *FrameRateIn*'in *FrameRateOut*'un bir katı olması gerekmektedir.

#### 3.2.1.4 FrameRateOut

*Double, default: 30*

İlgili katman için çıktı çerçeve hızını belirtir. Temel konfigürasyon dosyasındaki *FrameRate* parametresinin *FrameRateIn*'in bir katı ve *FrameRateIn*'in *FrameRateOut*'un bir katı olması gerekmektedir. *FrameRateOut*' un *FrameRateIn*' e eşit olmadığı durumlarda, girdi akışının her ( $\text{FrameRateIn} / \text{FrameRateOut}$ ). çerçevesi kodlanmadan atlanır. Bir katmanı kodlamak için kullanılan gerçek GOP büyüklüğü *FrameRate*,

*GOPSize*, ve *FrameRateOut* parametreleri vasıtasıyla belirlenir. Bir katman için kodlanacak olan gerçek çerçeve sayısı *FrameRate* ve *FrameRateOut* parametreleri vasıtasıyla belirlenir.

### 3.2.1.5 InputFile

*String, default: "test.yuv"*

Katman için orjinal işlenmemiş video akışının dosya adını belirtir.

### 3.2.1.6 ReconFile

*String, default: "rec.yuv"*

Katman için kodlanmış ve yeniden oluşturulmuş girdi akışının dosya adını belirtir. Bu akış, hata ayıklama (*debugging*) amaçlı kullanılmaktadır.

### 3.2.1.7 SymbolMode

*Flag (0 or 1), default: 1*

Entropi kodlama modunu belirtir. *SymbolMode* 0'a eşit iken, video akışı "variable length codes - VLC" kullanılarak kodlanır. *SymbolMode* 1'e eşit olduğunda, video akışı "context-adaptive binary arithmetic coding - CABAC" kullanılarak kodlanır. Genellikle CABAC, artırılmış kodlama etkinliği sağlamaktadır.

### 3.2.1.8 IDRPeriod

*Int, default: 0*

Katman içerisindeki IDR resimlerinin periyodunu belirtir. Olası değerler, N 0, 1, ... olmak üzere ( $GOPSize \times N$  şeklinde verilir).

## 3.2.2 Kodlama Parametreleri

### 3.2.2.1 ClosedLoop

*Unsigned Int, default: 0*

Kodlayıcının açık-döngü (open-loop) veya kapalı-döngü (closed-loop) modlarından hangisinde çalıştığını belirtir. Aşağıdaki değerler desteklenmektedir:

0, 1 – Tek-noktada kapalı-döngü kodlayıcı kontrolü

2 – İki-noktada kapalı-döngü kodlayıcı kontrolü

*ClosedLoop* 0 veya 1'e eşit olduğunda, tahmineleme döngüsü en büyük kalite noktasında kapanır. 2'ye eşit olduğunda, tahminleme döngüsü en düşük ve en yüksek kalite noktaları olmak üzere iki noktada kapanır. Eğer herhangi bir FGS katmanı mevcut değilse, *ClosedLoop* parametresinin 0, 1 ve 2 değerleri için aynı sonuçlar elde edilir. *ClosedLoop*'un 0 veya 1'e eşit olduğu mod ile karşılaştırmak gerekirse, FGS katmanları kodlandığında ve *ClosedLoop* 2'ye ayarlandığında, desteklenen en yüksek bit-hızında hız-bozulum (rate-distortion) etkinliği genellikle düşürülürken, desteklenen en düşük bit-hızında hız-bozulum (rate-distortion) etkinliği genellikle arttırılmış olur.

### 3.2.2.2 FRExt

*Flag (0 or 1), default: 0*

8x8 dönüşümün (High Profile) çalışır olup olmadığını belirtir. *FRExt* 1'e eşit iken, luminance bileşeni için 4x4 dönüşüme ek olarak 8x8 dönüşüm çalışır duruma getirilir; diğer durumlarda 8x8 dönüşüm etkisiz kılınır. Özellikle yüksek-çözünürlüklü kaynak materyali için, 8x8 dönüşüm çalışır duruma getirildiğinde kodlama etkinliği genellikle artmaktadır.

### 3.2.2.3 MaxDeltaQP

*Unsigned Int, default: 1*

Makroblok nicemleme parametrelerinin, resim nicemleme parametresinden maksimum mutlak farkını belirtir.

### 3.2.2.4 QP

*Double, default: 32.0*

İlgili katmanın kodlanması için temel nicemleme parametresini belirtir. Belirli bir çerçeveyi kodlamak için kullanılacak olan gerçek nicemleme parametreleri, bu *QP* parametresine ve çerçevenin GOP içerisindeki konumuna bağlı olarak belirlenir.

### 3.2.2.5 NumFGSLayers

*Double, default: 0*

İlgili uzamsal (spatial) veya kaba-taneli SNR katmanı içerisinde kodlanacak olan FGS katman sayısını belirtir. *NumFGSLayers* [0,3] aralığında olmalıdır. Kesirli bir rakam belirtildiğinde, son FGS katmanı, kodlama sembollerinin bu rakama karşılık gelen yüzdesinden sonra kesilir. Kesme oranı, bit-hızı oranı ile aynı değildir. Her FGS katmanı, nicemleme parametresinin 6 değeri düşmesine sebep olur.

### 3.2.2.6 FGSMotion

*Flag (0 or 1), default: 0*

FGS dilimlerinde hareket ayrıştırma yapılıp yapılmadığını belirtir. *FGSMotion* 0'a eşit iken, FGS dilimlerinden sadece dönüşüm katsayıları ayrıştırılır. *FGSMotion* 1'e eşit iken, bir makroblok için iletilen hareket ayrıştırma hız-bozulum eniyileştirmesi yoluyla belirlense bile, FGS dilimleri içerisindeki makrobloklar için yeni hareket verisinin iletilmesi ayrıca mümkün olmaktadır.

### 3.2.2.7 BaseLayerId

*Unsigned Int, default: bir sonraki alt katman*

Katmanlar-arası tahminleme için kullanılacak olan katmanı belirtir. *BaseLayerId* mevcut değilse, (temel konfigürasyon dosyasındaki yer alma sırası ile belirlenen) bir sonraki alt katmanın, katmanlar arası tahminleme için temel katmanı temsil ettiği düşünülür. Aksi takdirde (*BaseLayerId* mevcutsa), katmanlar-arası tahminleme için kullanılacak olan katman buna denk gelen *LayerId* ile belirlenir. *LayerCfg*'e bakınız.

### 3.2.2.8 ForceReOrdering

*Flag (0 veya 1), default: 0*

Referans resim listesinin her girdisi için yeniden düzenleme komutlarının kodlanıp kodlanmadığını belirtir. *ForceReOrdering* 0'a eşit olduğunda, sadece gerekli referans resim listesi yeniden düzenleme komutları iletilirken *ForceReOrdering* 1'e eşit olduğunda, referans resim listesindeki her girdi için bir yeniden düzenleme komutu iletilir. Yeniden düzenleme komutlarının kullanımının artması, kayıp resim tespit yeteneğini artırır.

### 3.2.2.9 EncSIPFile

*String, default: ""*

SIP karar dosyasının adını belirtir. Bu dosya, bu katmanın katmanlar-arası tahminleme kullanmayan çerçevelerinin POC'larının kaydını tutar. *EncSIPFile* mevcut değilse, kodlayıcı bu katman üzerinde SIP stratejisi kullanmaz.

## 3.2.3 Kontrol Parametreleri

### 3.2.3.1 MeQPLP

*Double, default: QP*

Anahtar resimlerin hareket tahminlemesi ve mod kararı için kullanılan *Lagrangian* parametresini belirtir. Mod kararı için *Lagrangian* parametresi  $L = 0.85 * (2^{(MeQPLP / 3)})$  şeklinde hesaplanır. Buna ek olarak, hareket tahminleme için *Lagrangian* parametresi, hareket arama için kullanılan bozulma ölçütüne bağlıdır. Bu parameter belirtilmediğinde, *QP* parametresine eşit olduğu sonucu çıkarılır.

### 3.2.3.2 MeQPX (X 0, 1, 2, 3, 4, veya 5 değerlerini alır)

*Double, default: 32.0*

Zamansal X katmanının hareket tahminlemesi ve mod kararı için kullanılan *Lagrangian* parametresini belirtir. Mod kararı için *Lagrangian* parametresi  $L = 0.85 * (2^{(MeQPX / 3)})$  şeklinde hesaplanır. Buna ek olarak, hareket tahminleme için *Lagrangian* parametresi, hareket aramam için kullanılan bozulma ölçütüne bağlıdır.

### 3.2.3.3 InterLayerPred

*Unsigned Int, default: 0*

Katmanlar-arası tahminleme kullanımını belirtir. Aşağıdaki değerler desteklenmektedir:

- 0 – Katman, katmanlar-arası tahminleme kullanmaz.
- 1 – Katmanlar-arası tahminleme her zaman kullanılır. Kullanmaz
- 2 – Katmanlar-arası tahminleme, macroblok uyumlu bir şekilde kullanılır.

Temel katman (LayerId 0) için *InterLayerPred* her zaman 0'a eşit olacaktır. Mod 1'de, kendisine karşılık gelen bir temel katman resmi var olduğu sürece, bir resmin tüm makroblokları için, katmanlar-arası tahminleme her zaman kullanılır duruma getirilir. Bu, özellikle hareket vektörlerinin her zaman temel katmandan kopyalandığı ve temel katmanın fark (*residual*) verisinin her zaman o anki farkların tahminlemesinde kullanıldığı anlamına gelmektedir. *InterLayerPred* 2'ye eşit iken, katmanlar-arası tahminleme araçları hız-bozulum eniyileştirme çatısı yoluyla rastgele seçilir. Ölçeklenebilir geliştirme katmanları için, en iyi kodlama etkinliği, *InterLayerPred* 2'ye ayarlandığında elde edilir.

### 3.2.3.4 BaseQuality

*Unsigned Int, default: 3*

Katmanlar-arası tahminleme için temel katmanın hangi kalite katmanını (*JSVM 6.8.2'de FGS katmanı, JSVM 9.12.2'de MGS katmanı*) kullanacağını belirtir. Bir uzamsal (*spatial*) veya kaba-taneli katman için 3'e kadar FGS katmanı mevcut olduğu için (*JSVM 9.12.2'e göre 16'ya kadar MGS katmanı mevcuttur*), *BaseQuality* [0,3] aralığında değer alacaktır (*JSVM 9.12.2'e göre 16'ya kadar MGS katmanı olduğu için [0,15] aralığında değer alacaktır*). *BaseQuality* parametresinin 0 olması, hiçbir FGS katmanının katmanlar-arası tahminlemede kullanılmadığı ve sadece temel gösterimin kullanıldığını belirtirken, bu parametrenin 1 değerini alması, buna ek olarak ilk FGS katmanının verisinin, ...vb kullanıldığı anlamına gelmektedir. 3 default değeri kullanıldığında, her durumda tüm temel katman verisi kullanılır.

## 3.2.4 Hareket Parametre Dosyası

### 3.2.4.1 MotionInfoMode

*Unsigned Int, default: 0*

Hareket verisinin dosyadan okunduğu veya dosyaya kaydedildiği bilgisini belirtir. Aşağıdaki değerler desteklenmektedir:

0 – hareket verisi, kaydedilmemiştir veya okunmamıştır.

1 – hareket verisi, *MotionInfoFile* tarafından belirtilen dosyadan okunmuştur.

2 – hareket verisi, *MotionInfoFile* tarafından belirtilen dosyaya kaydedilmiştir.

Burada, sadece *anahtar olmayan resimlerin* hareket verisinin bir hareket veri dosyasına kaydedildiği veya dosyadan okunduğuna dikkat edilmelidir. Kapalı-döngü (closed loop) kodlama gerçekleştirildiğinde ve nicemleme parametreleri değiştirildiğinde, doğrudan modda (direct mode) kodlanmış olan ve dosyadan okunan makroblok hareket verisi uygunsuz olabilir. Dosyadan hareket verisi okumak için bu olasılık kullanılırken özel bir itina göstermek gerekir.

### 3.2.4.2 MotionInfoFile

*String, default: "test.mot"*

*MotionInfoMode* parametresinde belirtildiği gibi, anahtar-olmayan resimlerin hareket verisinin okunması veya kaydedilmesi için kullanılan dosya adını belirtir.

## 3.2.5 Genişletilmiş Uzamsal Ölçeklenebilirlik Parametreleri

### 3.2.5.1 UseESS

*Int, default: 0*

Genişletilmiş Uzamsal Ölçeklenebilirliğin (*Extended Spatial Scalability - ESS*) kullanılıp kullanılmadığını belirtir. ESS, birbirini takip eden uzamsal katmanlar arasında genelleştirilmiş bir ilişkiye olanak tanır. Daha küçük uzamsal katmana ait bir resim, daha yüksek çözünürlükteki resmin kırılmış bir alanını temsil edebilir ve birbirini takip eden uzamsal katmanlar arasındaki oran ilişkisi ikili (ikinin katı şeklinde) olmak zorunda



değildir. Kırpma penceresini (*cropping window*) ve aşağı-örnekleme oranını tanımlayan geometrik parametreler dizi seviyesinde (*sequence level*) tanımlanabilir veya resim seviyesinde (*picture level*) geliştirilebilir. Aşağıdaki değerler desteklenmektedir:

- 0 – ESS kullanılmaz.
- 1 – Dizi seviyeli ESS
- 2 – Resim seviyeli ESS

### 3.2.5.2 ESSPicParamFile

*String, default: "ess.dat"*

Kırpma penceresi parametrelerini içeren dosya adını belirtir. Bu seçenek sadece, *UseESS* seçeneği 2'ye (Resim seviyeli ESS) ayarlandığında gereklidir. Dosyanın her satırı "*x\_orig, y\_orig, crop\_width, crop\_height*" virgüllerle ayrılmış dört tamsayıdan oluşturulur. Burada *x\_orig* ve *y\_orig* kırpma penceresinin üst sol köşesini temsil ederken, *crop\_width* ve *crop\_height* sırayla bu pencerenin genişliğini ve uzunluğunu temsil etmektedir. Kırpma penceresi parametreleri, yukarı örneklenmiş düşük katmanı temsil ediyor olsa da, daha yüksek katman alanında verilir.

### 3.2.5.3 ESSCropWidth

*Int, default: 0*

Kırpma penceresinin genişliğini belirtir. *UseESS* seçeneği 1 veya 2 ye ayarlandığında kullanılır.

### 3.2.5.4 ESSCropHeight

*Int, default: 0*

Kırpma penceresinin uzunluğunu belirtir. *UseESS* seçeneği 1 veya 2 ye ayarlandığında kullanılır.

### 3.2.5.5 ESSOriginX

*Int, default: 0*

Kırpma penceresinin üst sol köşesinin X-pozisyonunu belirtir. *UseESS* seçeneği 1'e ayarlandığında kullanılır.

### 3.2.5.6 ESSOriginY

*Int, default: 0*

Kırpma penceresinin üst sol köşesinin Y-pozisyonunu belirtir. *UseESS* seçeneği 1'e ayarlandığında kullanılır.

### 3.2.5.7 ESSChromaPhaseX

*Int, default: -1*

X-boyutunda yüksek (o anki) katman chroma bileşeninin X-boyutundaki luma bileşenine göre faz kaymasını belirtir.

### 3.2.5.8 ESSChromaPhaseY

*Int, default: 0*

X-boyutunda yüksek (o anki) katman chroma bileşeninin Y-boyutundaki luma bileşenine göre faz kaymasını belirtir.

### 3.2.5.9 ESSBaseChromaPhaseX

*Int, default: -1*

X-boyutunda daha düşük (temel) katman chroma bileşeninin X-boyutundaki luma bileşenine göre faz kaymasını belirtir.

### 3.2.5.10 ESSBaseChromaPhaseY

*Int, default: 0*

X-boyutunda daha düşük (temel) katman chroma bileşeninin Y-boyutundaki luma bileşenine göre faz kaymasını belirtir.

## 3.2.6 Uyarlanabilir-Referans FGS Parametreleri

### 3.2.6.1 EnhRefME

*Double, default: 0.0*

Hareket tahminleme için kullanılan referans çerçevesinin nasıl yapılandırıldığını belirtir. Esasında, bu parametre, temel katman (*base layer*) referans çerçevesi ve geliştirme katmanı (*enhancement layer*) referans çerçevesi arasında bir ağırlık çarpanı belirtir. Hareket tahminleme için referans çerçevesi;  $a$ , *EnhRefME* parametresi olmak üzere  $temel*(1-a)+enh*a$ ’a eşittir. Bu parameter için kabul edilebilir değer 0-1 aralığında olmalıdır. AR-FGS için genellikle 0.5 değeri seçilir.

### 3.2.6.2 WeightZeroBlock

*Unsigned Int, default: 32*

Anahtar resimlerin AR-FGS’i için kullanılan sıfır bloklar (*zero blocks*) için ağırlık çarpanı belirtir.

### 3.2.6.3 WeightZeroCoeff

*Unsigned Int, default: 32*

Anahtar resimlerin AR-FGS’i için kullanılan sıfır katsayıları (*zero coefficients*) için ağırlık çarpanı belirtir.

### 3.2.6.4 FgsEncStructure

*Unsigned Int, default: 1*

Anahtar resimlerin AR-FGS’i için kullanılan FGS sözdizim yapısını belirtir. *FGSEncStructure* 0’a eşit iken standart FGS sözdizimi kullanılır. *FGSEncStructure* 1’e eşit iken, eniyileştirilmiş FGS yapısı kullanılır.

### 3.2.6.5 FGSVectorMode

*Flag (0 veya 1), default: 0*

FGS katsayılarının vektör tarama modunu belirtir. 0 değeri, grupta büyüklüğünün belirtildiğini ve 1 değeri, bir vektörün belirtildiğini gösterir. Grupta büyüklüğü, her

dönüşteki (*cycle*) her blok için kodlanan FGS katsayı sayısını göstermektedir. Vektör, her dönüşte (*cycle*) her blok için erişilecek pozisyonu belirtir. Vektör, 4x4 bloklar için belirtilir ve 8x8 bloklar için vektör daha sonra çıkarsanır.

### 3.2.6.6 FGSGroupingSize

*Unsigned Int, default: 1*

Her dönüşte her blok için kodlanan FGS katsayıları grubunun büyüklüğünü belirtir.

### 3.2.6.7 FGSVectorX (X, 0-15 değerleri ile yer değiştirmektedir)

*Unsigned Int, default: 0*

X-inci vektörün pozisyonunu belirtir. Son pozisyon 15'e eşit olacaktır.

## 3.2.7 FMO ve Dilimler

### 3.2.7.1 SliceMode

*Unsigned Int, default: 0*

Dilim (*slice*) modunu ayarlar. Aşağıdaki değerler desteklenmektedir:

0-etkisiz kıl,

1-dilim başına sabit sayıda MB,

2- dilim başına sabit sayıda byte.

### 3.2.7.2 SliceArgument

*Unsigned Int, default: 50*

Mod 1 (MB sayısı) ve 2 (bayt sayısı) için dilim bağımsız değişkenlerini belirtir.

### 3.2.7.3 NumSlicGrpMns1

*Unsigned Int, default: 0*

Dilim grup sayısını belirtir. 0 değeri bir dilim grubu belirtirken, 1 değeri iki dilim grubu belirtir, ...vb.

### 3.2.7.4 SlcGrpMapType

*Unsigned Int, default: 2*

Dilim grup eşleme türünü belirtir. Aşağıdaki değerler desteklenmektedir:

0 – Interleaved Mod

1 – Dispersed Mod

2 – Foreground with left-over

3 – Box-out

4 – Raster scan

5 – Wipe

6 – Explicit Mod, *slice\_group\_id* *SlcGrpCfgFileNm* tarafından belirtilen dosyadan okunur.

### 3.2.7.5 SlcGrpChgDrFlag

*Unsigned Int, default: 0*

*slice\_group\_change\_direction\_flag*'ı ayarlar. Aşağıdaki değerler desteklenmektedir:

0 – Box-out clockwise, raster scan veya wipe right

1 – Box-out counter clockwise, reverse raster scan veya wipe left

### 3.2.7.6 SlcGrpChgRtMns1

*Unsigned Int, default: 85*

*slice\_group\_change\_rate\_minus1* sözdizim elemanının değerini belirtir.

### 3.2.7.7 SlcGrpCfgFileNm

*String, default: "sgcfg.cfg"*

0, 2 ve 6 dilim grubu eşleme (map) tipleri için kullanılan dilim konfigürasyon dosyasını belirtir. Eğer *NumSlicGrpMns1* > 0 , *SlcGrpMapType* = 0 ve *SliceMode* = 4 ise, tüm dilim gruplarının dizi uzunluğu (*runlength*) toplamı, (*SourceWidth*/16)'a eşit olmalıdır. IROI kodlama etkin hale getirilir ve IROI dilim bölme bilgisi kodlanır.

### 3.2.7.8 UseRedundantSlc

*Unsigned Int, default: 0*

Fazlalık olan dilimlerin kullanımını etkin hale getirir. Dilim başına bir fazlalık dilimi desteklemektedir. Geçerli fazlalık dilim aracı, resim başına çoklu dilim desteklememektedir.

### 3.2.7.9 PLR

*Unsigned Int, default: 0*

İlgili katman için paket kayıp hızını belirtir. Bu parametre, kayıp-farkındalıklı hız-bozulum eniyileştirme tarafından kullanılır, *LARDO*'ya bakınız.

### 3.2.7.10 NumROI

*Unsigned Int, default: 0*

İlgili katman için ROI sayısını belirtir.

### 3.2.7.11 ROICfgFileNm

*String, default "roiconf.cfg"*

ROI konfigürasyon dosya adını belirtir. *NumROI* > 0 ise, her ROI için *roi\_id*, *slice group id*, *scalability layer id* açıklanır.

## 4 Başlıca SVC kodlama parametrelerinin kodlama etkinliğine etkisi

Çalışma süresince, kodlayıcı parametreleri ve anlamları üzerine yapılan incelemeler sonucu, kodlama etkinliği üzerinde başlıca etkili parametrelerin;

- SNR ölçeklenebilirlik alternatifleri,
- GOP yapıları ve büyüklükleri,
- Maksimum yapısal gecikme sınırlandırması,
- Referans çerçeve sayıları,
- Hareket tahminleme parametreleri,

şeklinde listelenebileceği kanaatine varıldı. Yine çalışma dahilinde bu parametrelerden bazılarının kodlama kalitesine etkileri gözlemlenerek yorumlamaları gerçekleştirildi. Bu bölümde, yapılan testler sonucu elde edilen edinimler görsel olarak sunulmakta ve kısaca yorumlanmaktadır.

Tüm testler 30 Hz. qcif Foreman ve Coastguard videoları üzerinde gerçekleştirilmiştir. Test videolarının CIF çözünürlüğünde seçilmesinin, elde edilen sonuçları daha belirgin olarak göstereceği bilinmektedir; ancak kodlama sürelerinin çok uzayacağı düşünüldüğünden testler için QCIF çözünürlükler tercih edilmiştir. Test örneklerinin oluşturulmasında [8, 9, 10] nolu kaynaklardan faydalanılmıştır.

### 4.1 Farklı SNR alternatiflerinin kodlamaya etkileri:

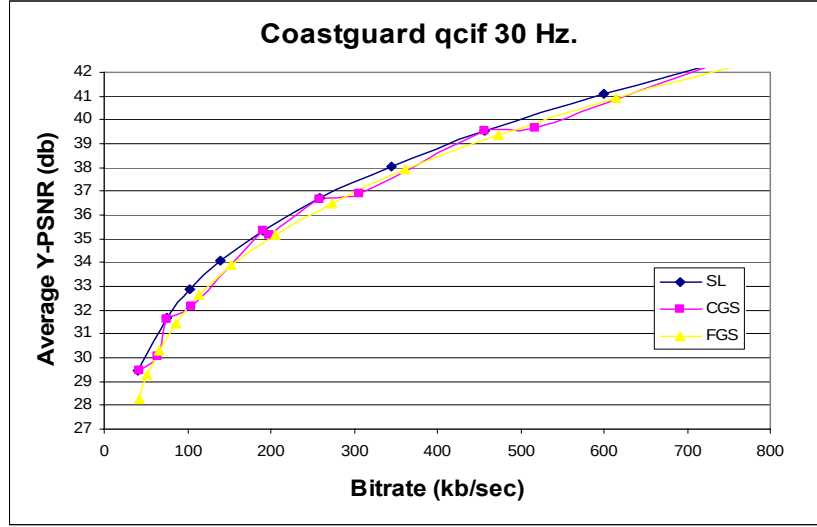
Farklı SNR ölçeklenebilirlik alternatiflerinin kodlama etkinliğine katkılarını görebilmek amacıyla, tek katman (*single layer – SL*), CGS ve FGS seçeneklerinin benzer kalite değerleri için kodlanan videonun bit hızını (*bit rate*) nasıl etkilediğine dair ölçümler yapılmıştır. İki farklı video üzerinde yapılan bu ölçümlerin sonuçları Grafik 1 ve 2’de görülebilmektedir. Tablo 1 ve 2’den bu ölçümlere ait sayısal değerlere ulaşılabilmektedir.

SNR ölçeklenebilirlik sağlamak için kullanılan tüm ölçümlerde  $GOPSize = 16$  ve  $IntraPeriod = -1$  seçmek yoluyla IBBBBBBBBBBBBBBBP GOP yapısı (*hierarchical B pictures*) kullanılmıştır.

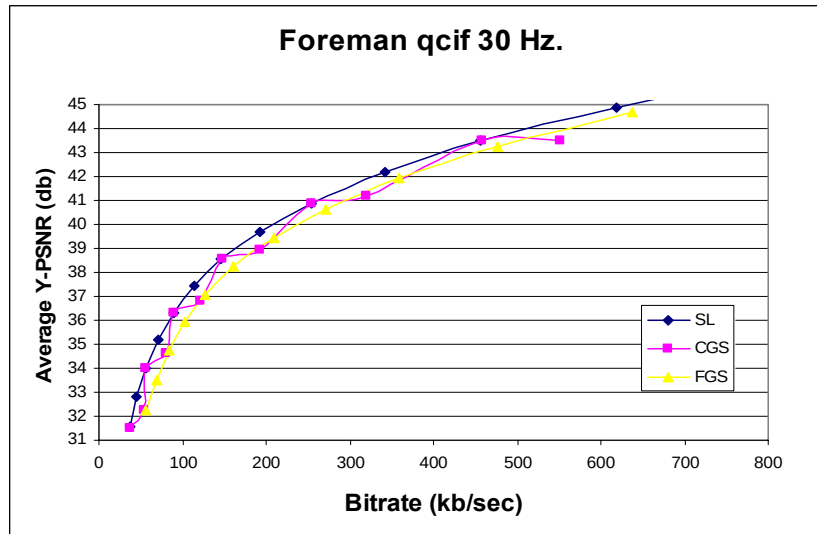
Tablo 3 ve 4 ve Grafik 1 ve 2’den de görülebildiği gibi aynı kalitede bir video üç farklı yöntemle kodlandığında, SL yöntemiyle en düşük bit hızı elde edilmektedir. Sırasıyla CGS ve FGS daha yüksek bit hızlarını sunmaktadır. Aynı bit hızında SL ve CGS ile birbirine yakın görüntü kalitesi elde edilirken, FGS ile aynı bit hızında elde edilen görüntü kalitesi bir miktar daha düşüktür. Bu kötü getirime rağmen FGS’nin tercih edilmesini sağlayan sebep, seçilen qp değeri ile yapılacak tek bir kodlamada elde edilecek videonun, geniş bir bit hızı aralığında istenen bit hızında kod çözümü yapılmasının mümkün olmasıdır. Bu sayede FGS kodlanmış videonun aktarımı sırasında oluşmuş olabilecek kayıplar sebebiyle kod çözümü sırasında elde edilen bit hızında çok büyük düşüşler gözlenmeyecektir. CGS yöntemi ile böyle bir seçim şansı yoktur. CGS kodlamada her kalite katmanı belirtilen qp değeri ile kodlanır ve her bir katmanın kod çözümü sonucunda tek bir bit hızı elde edilebilir. Bunun yanı sıra, FGS yöntemi

kullanıldığında seçilen bir nicemleme parametresi ile elde edilen görüntü kalitesi, CGS ve SL yöntemleri **aynı parametre** ile çalıştırıldığında elde edilenden daha yüksektir (bkz. Tablo 3 ve 4).

Bu çalışma dahilinde yapılan CGS ölçümlerinde  $\Delta QP=2$  değeri kullanılmıştır. FGS kullanan kodlama işlemlerinde tek katman kullanılmıştır, bu sebeple  $\Delta QP=6$  olmaktadır. FGS yöntemi uygulanırken herhangi bir iyileştirme opsiyonu kullanılmamıştır; bu sebeple elde edilen sonuçların CGS ile yakın olduğu gözlenmektedir. Ancak özellikle büyük bit hızı aralıkları hedeflendiğinde FGS yöntemini hareket ayırıştırma (*motion refinement*) opsiyonu ile birlikte kullanmak doğru bir seçim olacaktır. CGS yöntemi kullanılarak yapılan kodlamalarda  $\Delta QP$  değeri arttıkça kodlama etkinliğinin arttığı bilinmektedir.



**Grafik 1.** SL, CGS ve FGS SNR ölçeklenebilirlik alternatiflerinin Coastguard videosunun kodlanma etkinliği üzerindeki etkileri.



**Grafik 2.** SL, CGS ve FGS SNR ölçeklenebilirlik alternatiflerinin Foreman videosunun kodlanma etkinliği üzerindeki etkileri.

| Single Layer |                |             | CGS |                |             | FGS |                |             |
|--------------|----------------|-------------|-----|----------------|-------------|-----|----------------|-------------|
| QP           | Bitrate (kbps) | Y-PSNR (db) | QP  | Bitrate (kbps) | Y-PSNR (db) | QP  | Bitrate (kbps) | Y-PSNR (db) |
| 20           | 769,1008       | 42,6837     | 18  | 850,6384       | 42,6529     | 22  | 1004,928       | 44,1683     |
| 22           | 599,9456       | 41,1009     | 20  | 770,9392       | 42,6954     | 24  | 791,2816       | 42,5124     |
| 24           | 456,4288       | 39,5342     | 22  | 518,0672       | 39,6753     | 26  | 615,0288       | 40,9085     |
| 26           | 344,5216       | 38,0677     | 24  | 457,0384       | 39,5382     | 28  | 474,288        | 39,3758     |
| 28           | 258,5264       | 36,6901     | 26  | 305,5824       | 36,8978     | 30  | 361,6816       | 37,906      |
| 30           | 190,6992       | 35,3334     | 28  | 258,7072       | 36,6802     | 32  | 272,936        | 36,5057     |
| 32           | 139,2192       | 34,0551     | 30  | 198,896        | 35,1657     | 34  | 205,2352       | 35,1544     |
| 34           | 102,6656       | 32,8571     | 32  | 190,9968       | 35,3289     | 36  | 152,656        | 33,8787     |
| 36           | 74,9952        | 31,6662     | 34  | 105,4432       | 32,1622     | 38  | 114,6224       | 32,6549     |
| 40           | 41,0496        | 29,4782     | 36  | 74,6544        | 31,6371     | 40  | 86,2432        | 31,4631     |
|              |                |             | 38  | 64,0736        | 30,0788     | 42  | 66,5488        | 30,3278     |
|              |                |             | 40  | 41,592         | 29,4649     | 44  | 52,272         | 29,3038     |
|              |                |             |     |                |             | 46  | 41,3536        | 28,2892     |

**Tablo 3.** 30Hz. qcif Coastguard videosunda, SNR ölçeklenebilirlik sağlamak için kullanılan farklı alternatiflerin kodlama etkinliğine katkıları.

| Single Layer |                |             | CGS |                |             | FGS |                |             |
|--------------|----------------|-------------|-----|----------------|-------------|-----|----------------|-------------|
| QP           | Bitrate (kbps) | Y-PSNR (db) | QP  | Bitrate (kbps) | Y-PSNR (db) | QP  | Bitrate (kbps) | Y-PSNR (db) |
| 16           | 825,0304       | 46,345      | 18  | 551,0368       | 43,5299     | 22  | 638,1344       | 44,6673     |
| 18           | 618,2704       | 44,8848     | 20  | 457,4624       | 43,5299     | 24  | 476,3344       | 43,251      |
| 20           | 456,832        | 43,4708     | 22  | 320,0528       | 41,1689     | 26  | 359,1088       | 41,9171     |
| 22           | 341,448        | 42,1657     | 24  | 254,9072       | 40,8799     | 28  | 270,5056       | 40,6133     |
| 24           | 254,0864       | 40,8778     | 26  | 192,368        | 38,9426     | 30  | 208,5408       | 39,4108     |
| 26           | 192,0576       | 39,7024     | 28  | 147,7232       | 38,5603     | 32  | 161,32         | 38,227      |
| 28           | 146,6672       | 38,5606     | 30  | 121,0576       | 36,7981     | 34  | 127,92         | 37,0593     |
| 30           | 114,1344       | 37,4354     | 32  | 88,8336        | 36,3072     | 36  | 102,0432       | 35,9077     |
| 32           | 89,3888        | 36,3173     | 34  | 80             | 34,6257     | 38  | 83,8768        | 34,7426     |
| 34           | 70,7936        | 35,1789     | 36  | 56,7344        | 33,9818     | 40  | 68,7616        | 33,4717     |
| 36           | 56,696         | 33,9857     | 38  | 54,7392        | 32,247      | 42  | 56,4336        | 32,2514     |
| 38           | 45,6992        | 32,7969     | 40  | 37,3152        | 31,5167     |     |                |             |
| 40           | 36,9488        | 31,5458     |     |                |             |     |                |             |

**Tablo 4.** 30Hz. qcif Foreman videosunda, SNR ölçeklenebilirlik sağlamak için kullanılan farklı alternatiflerin kodlama etkinliğine katkıları.

## 4.2 Farklı tahminleme yapılarının kodlamaya etkileri:

Yapılandırma parametrelerinin kodlamaya etkilerini incelemek üzere ikili hierarşik tahminleme (*dyadic hierarchical prediction*) yapıları ile konvensiyonel tahminleme (*conventional prediction*) yapılarının karşılaştırılması yapılmıştır. Konvensiyonel tahminleme yapıları arasından IPPP ve IBPBP yapıları, ikili hiyerarşik tahminleme yapıları arasından da GOP büyüklüğü 4 ve 8 ve IntraPeriod değeri -1 ve 4 olan üç farklı yapı seçilmiştir. Kullanılan GOP yapıları ve bu yapıları oluşturmak için seçilen parametre kombinasyonları Tablo5'te görüldüğü gibidir.

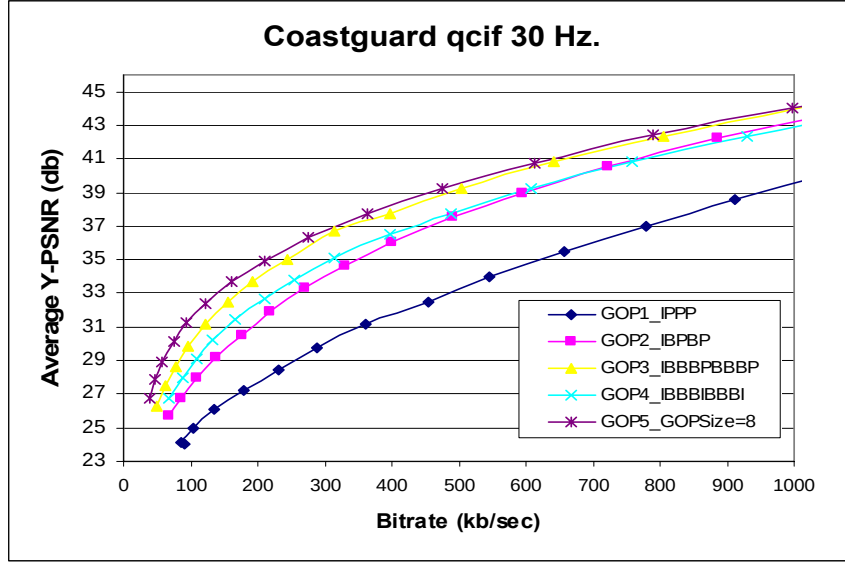
|             | GOP1                | GOP2                | GOP3                | GOP4                | GOP5                |
|-------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| GOP yapısı  | IPPP                | IBPBP               | IBBBPBBBBP          | IBBBIBBBI           | IBBBBBBBP           |
| GOPSize     | 1                   | 2                   | 4                   | 4                   | 8                   |
| IntraPeriod | -1                  | -1                  | -1                  | 4                   | -1                  |
| NumFGSLayer | 1 ( $\Delta QP=6$ ) | 1 ( $\Delta QP=6$ ) | 1 ( $\Delta QP=6$ ) | 1 ( $\Delta QP=6$ ) | 1 ( $\Delta QP=6$ ) |
| SymbolMode  | 1 (CABAC)           | 1 (CABAC)           | 1 (CABAC)           | 1 (CABAC)           | 1 (CABAC)           |

**Tablo 5.** Farklı GOP yapılarının video kodlanma etkinliği üzerindeki etkilerini ölçmek amacıyla kullanılan tahminleme yapıları ve seçilen parametre değerleri.

Testlerde öncelikle en basit konvensiyonel tahminleme yapısına yer vermek istenmiş ve bunun için GOP1 yapısı seçilmiştir. GOP1'de herhangi bir B-çerçeve bulunmamakta ve sadece ileriye dönük referans yapısı kullanılarak tahminleme yapılmaktadır. İkinci konvensiyonel tahminleme yapısı olarak GOP2 seçilerek, B tipi çerçeve kullanımının P tipi çerçeve kullanımına (GOP1 ile karşılaştırıldığında) göre görüntü kalitesi ve bit hızı açısından sağladığı üstünlük gözlenmek istenmiştir. GOP3'de kullanılan B tipi çerçeve sayısı GOP2'ye göre daha fazladır ve GOP3'de kodlama karmaşıklığı yüksek olsa dahi, aynı kalitede kodlanmış görüntü elde etmek için daha az veri kullanılacaktır. İkili hierarşik tahminleme yapıları arasında GOP4 yapısının seçilme sebebi, daha fazla sayıda I tipi çerçeve içermesinin diğer yapılara göre daha fazla hataya karşı dayanıklılık (*error resilience*) imkanı sağlamasıdır. Artan GOP büyüklüklerinin kodlamaya etkilerini görebilmek adına ise GOP5 yapısı test durumları içine dahil edilmiştir. Her GOP yapısı için farklı qp değerleri ile kodlamalar yapılmış ve elde edilen bit hızları ve onlara karşılık gelen PSNR değerleri yorumlanmıştır. Testlerin sayısal değerleri Tablo 6 ve 7'de, görsel olarak temsil edilmiş test sonuçları ise Grafik 3 ve 4'te görülmektedir.

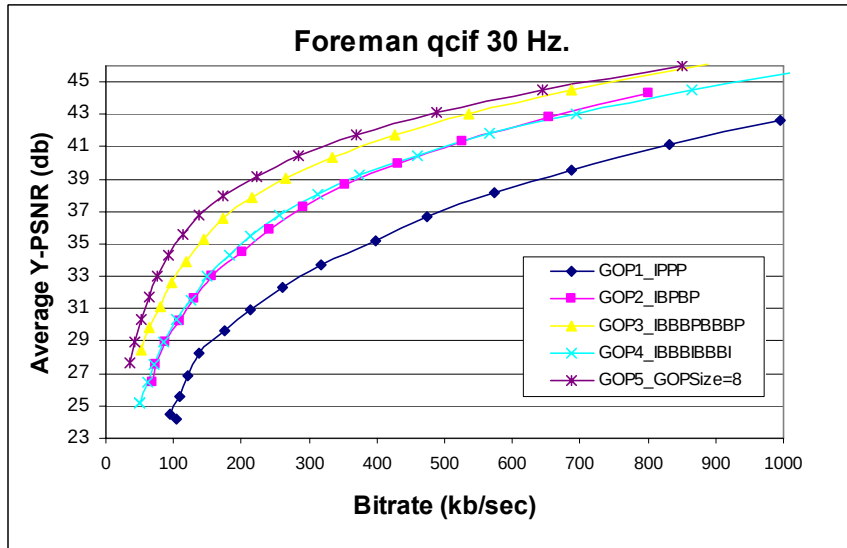
Grafik 3 ve 4'den de görülebileceği gibi ikili hiyerarşik tahminleme yapıları, konvensiyonel alternatiflerine göre daha etkin kodlama yapılmasını sağlamıştır. GOP yapısının büyüklüğü arttıkça birim bit hızı ile temsil edilen video kalitesi de artış göstermektedir. Özellikle B tipi çerçevelerin kullanımının kodlama etkinliğine katkısı büyüktür. Bu katkının en rahatlıkla gözlemlenebildiği ölçüm GOP1 ve GOP2 eğrileri arasındaki farktır. Bu yapılarda B tipi çerçeve kullanımıyla ortalama 3 db.lik bir kalite artışı yakalanmıştır. Kodlama etkinliği GOP büyüklüğüne bağlı olarak artış göstermektedir. Orta seviyelerdeki bit hızlarında GOP büyüklüğü artışıyla birlikte ortalama 1 db.lik kalite artışı yakalanmıştır.





**Grafik 3.** Farklı GOP yapılarının Coastguard videosunun kodlanma etkinliği üzerindeki etkileri.

GOP3 ve GOP4 tahminleme yapıları karşılaştırıldığında I tipi çerçeve kullanımı ile P tipi çerçeve kullanımının görüntü kalitesi ve bit hızına etkileri açıkça görülmektedir. Aynı nicemleme parametresi ile kodlama yapıldığında, I tipi çerçeve kullanımını sıklaştıran tahminleme yapısı ile daha yüksek görüntü kalitesi elde edilirken yine daha yüksek bit hızı da ortaya çıkmaktadır.



**Grafik 4.** Farklı GOP yapılarının Foreman videosunun kodlanma etkinliği üzerindeki etkileri.

İki yönlü (*bi-directional*) tahminlemede, kodlayıcı, kendisinden önceki ve sonraki referans çerçevelerine dayalı tahminleme yaparken, kendisinden önceki, sonraki ve her ikisinin ortalamasına karşılık gelen çerçeveler arasında en iyi eşleşme sonucu veren hareket vektörünü

bulmaya çalışmaktadır. Bu işlemin kodlama karmaşıklığı, P tipi çerçeve ve I tipi çerçeve kodlamaya göre oldukça fazladır. İki yönlü tahminlemeden daha az kodlama karmaşıklığı kullanan P tipi çerçeve kodlama, sadece kendisinden önce gelen referans çerçevelerine dayalı tahminleme yapmaktadır. Herhangi bir referans çerçeve kullanmayan I tipi çerçeve kodlamanın kodlama karmaşıklığı en küçüktür. Kodlama karmaşıklığı arttıkça, kodlama süresi artsa dahi, kodlama için kullanılan veri miktarı azalacağından bit hızı da küçülecektir. Sonuç olarak, kodlama karmaşıklıkları  $B > P > I$  şeklinde sıralanırken bit hızını artırma etkisi tam tersi şekilde  $I > B > P$  sıralamasına sahiptir. Grafik 3 ve 4 incelendiğinde, GOP2 ve GOP4'e ait eğrilerin kimi zaman yer değiştirerek birbirine çok yakın seyrediyor olması bu durumla açıklanabilmektedir.

Kodlama etkinliğinin farklı GOP yapılarına tepkisinin ölçüldüğü bu testlerde tek katmanlı ( $\Delta QP=6$ ) FGS SNR ölçeklenebilirlik özelliği kullanılmıştır. Bu test ölçümleri sırasında yapısal gecikme (*structural delay*) değeri için herhangi bir sınırlandırma getirilmemiştir. Sınırlandırma söz konusu olduğu durumlarda belirli bir GOP büyüklüğüne kadar kodlama etkinliği artış göstermekte ancak referans yapısı gecikme sınırına ulaştıktan sonra azalan bir seyir izlemektedir. Yapısal gecikmenin etkileri bir sonraki test grubunda ele alınmıştır.

| GOP1 |                |             | GOP2 |                |             | GOP3 |                |             | GOP4 |                |             | GOP5 |                |             |
|------|----------------|-------------|------|----------------|-------------|------|----------------|-------------|------|----------------|-------------|------|----------------|-------------|
| QP   | Bitrate (kbps) | Y-PSNR (db) | QP   | Bitrate (kbps) | Y-PSNR (db) | QP   | Bitrate (kbps) | Y-PSNR (db) | QP   | Bitrate (kbps) | Y-PSNR (db) | QP   | Bitrate (kbps) | Y-PSNR (db) |
| 20   | 1648,3152      | 45,7317     | 20   | 1285,43        | 45,491      | 20   | 1250           | 45,8        | 20   | 1398,96        | 45,7635     | 20   | 1244           | 45,7278     |
| 22   | 1422,1168      | 43,7256     | 22   | 1070,24        | 43,792      | 22   | 1008           | 44          | 22   | 1146,15        | 44,03       | 22   | 997,8          | 44,0369     |
| 24   | 1241,4576      | 41,9993     | 24   | 885,571        | 42,217      | 24   | 805,1          | 42,3        | 24   | 931,134        | 42,3312     | 24   | 788,3          | 42,389      |
| 26   | 1068,5072      | 40,2666     | 26   | 721,667        | 40,511      | 26   | 642,8          | 40,8        | 26   | 758,242        | 40,8205     | 26   | 612,8          | 40,7362     |
| 28   | 911,0288       | 38,5443     | 28   | 593,706        | 38,933      | 28   | 504,7          | 39,2        | 28   | 608,133        | 39,2662     | 28   | 474,5          | 39,2059     |
| 30   | 780,216        | 36,9956     | 30   | 489,84         | 37,554      | 30   | 396,5          | 37,7        | 30   | 488,925        | 37,7793     | 30   | 363,7          | 37,7328     |
| 32   | 657,96         | 35,5224     | 32   | 399,435        | 36,029      | 32   | 313,7          | 36,7        | 32   | 396,386        | 36,4843     | 32   | 276            | 36,2931     |
| 34   | 544,6672       | 33,9756     | 34   | 329,067        | 34,643      | 34   | 245,3          | 35          | 34   | 315,461        | 35,0681     | 34   | 209,6          | 34,9449     |
| 36   | 453,9424       | 32,5027     | 36   | 270,632        | 33,3        | 36   | 193,4          | 33,7        | 36   | 253,893        | 33,8261     | 36   | 160,1          | 33,6904     |
| 38   | 362,2544       | 31,1463     | 38   | 218,085        | 31,874      | 38   | 155,9          | 32,5        | 38   | 209,186        | 32,6488     | 38   | 121,2          | 32,4255     |
| 40   | 287,2448       | 29,755      | 40   | 175,619        | 30,524      | 40   | 122,6          | 31,2        | 40   | 166,576        | 31,451      | 40   | 94,43          | 31,2209     |
| 42   | 230,0768       | 28,4673     | 42   | 137,318        | 29,223      | 42   | 97,13          | 29,8        | 42   | 133,131        | 30,2277     | 42   | 74,36          | 30,0978     |
| 44   | 178,5136       | 27,2662     | 44   | 108,354        | 27,93       | 44   | 78,56          | 28,7        | 44   | 110,027        | 29,1458     | 44   | 58,43          | 28,9535     |
| 46   | 135,3744       | 26,0972     | 46   | 86,2032        | 26,715      | 46   | 61,56          | 27,5        | 46   | 87,024         | 27,9383     | 46   | 47,97          | 27,8562     |
| 48   | 104,936        | 24,9525     | 48   | 67,2512        | 25,689      | 48   | 50,3           | 26,3        | 48   | 68,728         | 26,7608     | 48   | 38,45          | 26,7661     |
| 50   | 86,5472        | 24,1726     |      |                |             |      |                |             |      |                |             |      |                |             |
| 52   | 91,128         | 23,9923     |      |                |             |      |                |             |      |                |             |      |                |             |

**Tablo 6.** 30Hz. qcif Coastguard videosunda, farklı taminleme yapılarının kodlamaya etkisini gözlemlemek için yapılan testlere ait sayısal değerler.

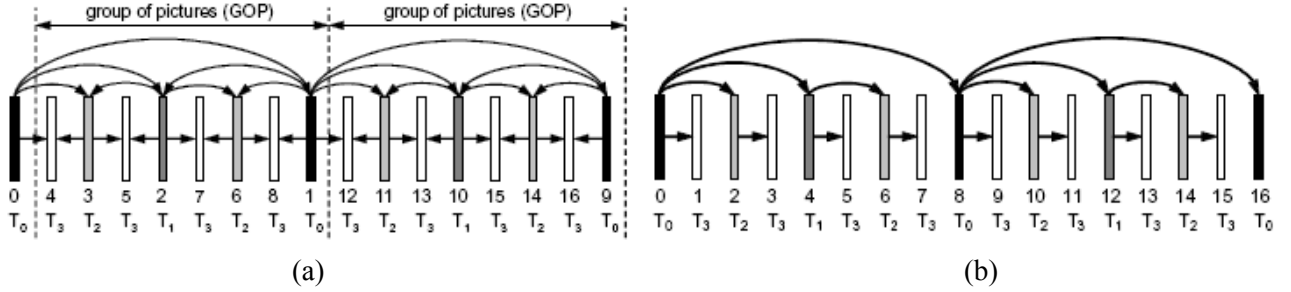
| GOP1 |                |             | GOP2 |                |             | GOP3 |                |             | GOP4 |                |             | GOP5 |                |             |
|------|----------------|-------------|------|----------------|-------------|------|----------------|-------------|------|----------------|-------------|------|----------------|-------------|
| QP   | Bitrate (kbps) | Y-PSNR (db) | QP   | Bitrate (kbps) | Y-PSNR (db) | QP   | Bitrate (kbps) | Y-PSNR (db) | QP   | Bitrate (kbps) | Y-PSNR (db) | QP   | Bitrate (kbps) | Y-PSNR (db) |
| 24   | 994,5184       | 42,673      | 22   | 800,9936       | 44,2766     | 20   | 889,536        | 46,0846     | 20   | 1086,608       | 46,098      | 20   | 851,5376       | 46,0338     |
| 26   | 831,8048       | 41,0942     | 24   | 654,3552       | 42,8485     | 22   | 687,6512       | 44,5183     | 22   | 864,808        | 44,5391     | 22   | 645,0624       | 44,5361     |
| 28   | 686,6592       | 39,5726     | 26   | 525,5024       | 41,3597     | 24   | 535,984        | 43,0556     | 24   | 693,3984       | 43,0641     | 24   | 487,4192       | 43,1202     |
| 30   | 572,584        | 38,1271     | 28   | 431,3744       | 39,9557     | 26   | 426,792        | 41,7572     | 26   | 566,4576       | 41,7898     | 26   | 368,8864       | 41,7001     |
| 32   | 473,2112       | 36,7013     | 30   | 354,176        | 38,6551     | 28   | 335,2016       | 40,3836     | 28   | 459,0496       | 40,4899     | 28   | 284,9808       | 40,433      |
| 34   | 398,1504       | 35,1897     | 32   | 291,3568       | 37,2718     | 30   | 266,4736       | 39,0705     | 30   | 374,2576       | 39,2112     | 30   | 222,1552       | 39,1831     |
| 36   | 317,9616       | 33,6748     | 34   | 241,7488       | 35,9238     | 32   | 215,7552       | 37,8379     | 32   | 312,4976       | 38,0755     | 32   | 173,632        | 37,9671     |
| 38   | 260,616        | 32,3646     | 36   | 200,2688       | 34,516      | 34   | 173,768        | 36,5517     | 34   | 256,0032       | 36,7669     | 34   | 138,1536       | 36,762      |
| 40   | 213,5696       | 30,9314     | 38   | 156,0944       | 33,0526     | 36   | 144,6944       | 35,2601     | 36   | 213,8944       | 35,5379     | 36   | 113,3184       | 35,5836     |
| 42   | 175,752        | 29,5997     | 40   | 129,464        | 31,6408     | 38   | 117,3696       | 33,9449     | 38   | 182,9184       | 34,3377     | 38   | 93,3728        | 34,2973     |
| 44   | 137,7216       | 28,222      | 42   | 108,2864       | 30,2738     | 40   | 97,0496        | 32,5795     | 40   | 150,4512       | 32,9968     | 40   | 76,3456        | 33,0431     |
| 46   | 121,928        | 26,8819     | 44   | 87,8576        | 28,9573     | 42   | 79,9504        | 31,1366     | 42   | 124,8928       | 31,5559     | 42   | 63,0304        | 31,7664     |
| 48   | 109,6816       | 25,5706     | 46   | 72,9728        | 27,5663     | 44   | 63,9648        | 29,8147     | 44   | 105,408        | 30,3324     | 44   | 51,9936        | 30,3073     |
| 50   | 95,912         | 24,5117     | 48   | 68,1168        | 26,433      | 46   | 53,216         | 28,4746     | 46   | 86,4704        | 28,9141     | 46   | 43,6288        | 28,9967     |
| 52   | 104,872        | 24,2384     |      |                |             |      |                |             | 48   | 71,4672        | 27,5576     | 48   | 36,2048        | 27,6134     |
|      |                |             |      |                |             |      |                |             | 50   | 62,0016        | 26,43       |      |                |             |
|      |                |             |      |                |             |      |                |             | 52   | 50,1056        | 25,2146     |      |                |             |
|      |                |             |      |                |             |      |                |             | 54   | 50,6992        | 25,1361     |      |                |             |

**Tablo 7.** 30Hz. qcif Foreman videosunda, farklı taminleme yapılarının kodlamaya etkisini gözlemlemek için yapılan testlere ait sayısal değerler.

### 4.3 Maksimum yapısal gecikmenin farklı GOP büyüklüklerinde kodlama kalitesine etkileri:

Bir önceki bölümde anlatılan testlerde tahminleme yapılarının kodlama etkinliğine katkılarını incelerken, herhangi bir yapısal gecikme kısıtı göz önünde bulundurulmamıştır. Ancak, GOP büyüklüğü arttıkça ve dolayısıyla kullanılan B tipi çerçevelerin sayısı çoğaldıkça karşılaşılan yapısal gecikmenin de artış göstereceği bilinen bir gerçektir.

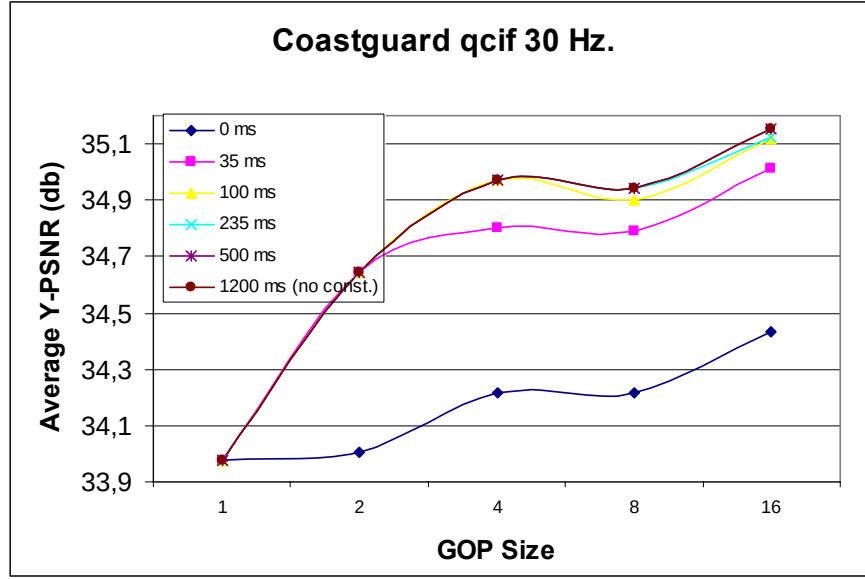
Yapısal gecikme, GOP yapısından kaynaklanan geriye dönük referans mekanizması sebebiyle kodlama sırasında karşılaşılan gecikmedir. Bu gecikme çerçeve sayısı ya da ona karşılık gelen süre cinsinden temsil edilebilmektedir. Çerçeve sayısı bazında açıklanacak olursa, kod çözücü tarafında karşılaşılan gecikme, kodlanmış ancak çıktı olarak gönderilmeden DPB'da tutulması gereken çerçeve sayısı olarak düşünülebilir. Benzer şekilde, kodlayıcı tarafında da kodlanmış ancak kendisinden sonraki çerçevelere referans olarak kullanılmak üzere tamponda (*buffer*) tutulması gereken maksimum çerçeve sayısıdır. Dolayısıyla çerçeve bazında kodlayıcı gecikmesi geriye dönük tahminleme paterninde ilk ve son çerçevelerin gösterim sıraları arasındaki maksimum farktır [8]. Şekil 7 incelenecek olursa, 7.a'da görülen tahminleme yapısının gecikme miktarı 8 çerçeve olurken 7.b'deki yapının gecikme miktarı sıfır çerçevedir.



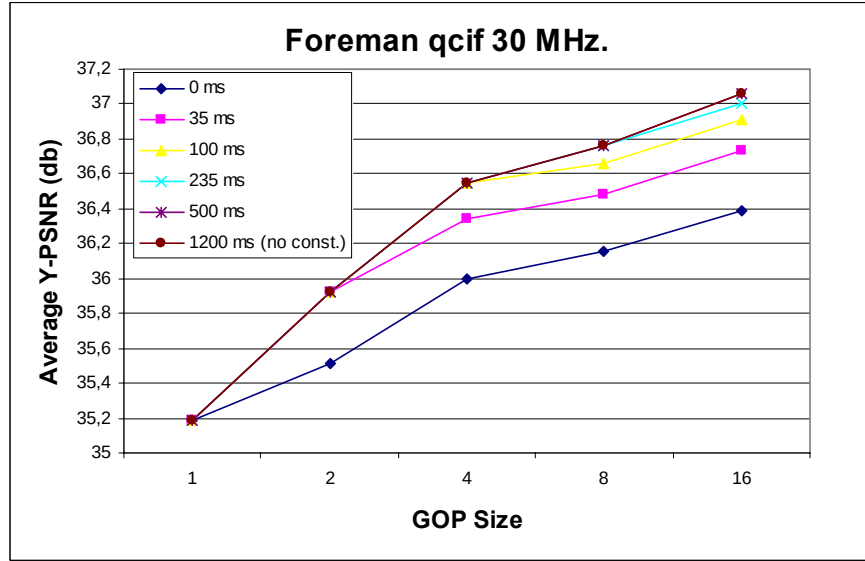
Şekil 7. Zamansal ölçeklenebilirlik sağlamak için kullanılan hiyerarşik tahminleme yapılarının yapısal gecikme özellikleri [8].

Kodlama sırasında yapısal gecikme miktarı aksi belirtilmediği sürece kısıtlandırılmamaktadır; ancak gecikmeye duyarlı çalışmalarda ihtiyaca göre düzenlenmelidir. Bu test grubunda amacımız farklı GOP uzunluklarında kodlanmış videoların farklı maksimum yapısal gecikme kısıtı değerlerine verdiği tepkiyi gözlemlemektir. Bu amaçla ikili hiyerarşik tahminleme yapıları değişik GOP büyüklükleri için test konusu yapılmıştır. Yapılan testlerde, tek katmanlı ( $\Delta QP=6$ ) FGS SNR ölçekleme yöntemi kullanılmış, nicemleme parametresi  $QP=34$  ve  $IntraPeriod=-1$  değerlerinde sabit tutulmuştur. Grafik 5 ve 6'da bu testin sonuçları görülebilmektedir.

Test sonuçlarından da görülebildiği gibi, GOP büyüklüğündeki artışa bağlı olarak oluşan yapısal gecikme, belirlenen kısıt değere ulaştığında, elde edilen kodlama kalitesi düşüş göstermektedir. Bunun sebebi, belirlenen gecikme kısıt değerine sadık kalabilmek için tahminleme yapısının değiştirilerek kodlama kalitesinin bilinçli olarak düşürülmesidir. Tahmin edilebileceği gibi, GOP büyüklükleri arttırıldıkça sonuç grafiklerinde görülen yapı aynen korunacak ve yüksek gecikme kısıtları için de kalite düşüşleri görülecektir.



**Grafik 5.** Farklı GOP büyüklüklerinin değişik yapısal gecikme kısıtlarına verdiği tepkinin Coastguard videosunun kodlanma etkinliği üzerindeki etkileri.



**Grafik 6.** Farklı GOP büyüklüklerinin değişik yapısal gecikme kısıtlarına verdiği tepkinin Foreman videosunun kodlanma etkinliği üzerindeki etkileri.

## 5 Sonuç

Bu çalışmada, H.264 Ölçeklenebilir Video Kodlayıcı (SVC)'sının kodlama yapısına genel olarak değinilmiş ve kodlama parametreleri detaylı olarak incelenmiştir. Tüm parametrelerin kodlama aşamasındaki anlamları ve alabildikleri değerlerin anlamları yorumlanmıştır. Kodlama aşaması için en çok önem taşıyan parametreler listelenmiş ve bu parametrelerden bazılarının kodlama üzerindeki etkilerinin gözlemlenmesi amacıyla kodlama testleri gerçekleştirilmiştir.

## 6 Kaynaklar

[1] Gül Boztok Algın, 2007, "Ölçeklenebilir H.264 video kodlayıcısı için seviyelendirilebilir güvenlik sağlayan bir video şifreleme çalışması, Ege Üniversitesi Yüksek Lisans Tezi.

[2] H. Schwarz, D. Marpe, T. Wiegand, 2006, "The Scalable H.264/MPEG4-AVC Extension - Technology and Applications,, , Proc. European Symposium on Mobile Media Delivery (EuMob 2006), Alghero, Sardinia, Italy.

[3] Heinrich Hertz Institut (HHI), Image Communication Research Projects: The Scalable Video Coding Amendment of the H.264/AVC Standard, [http://ip.hhi.de/imagecom\\_G1/savce/](http://ip.hhi.de/imagecom_G1/savce/)

[4] Heiko Schwarz, Mathias Wien, 2008, "The Scalable Video Coding Extension of the H.264/AVC Standard {Standards in a nutshell}", IEEE Signal Processing Magazine, Vol. 25, No. 2, pp. 135-141, Mar. 2008, Invited Paper.

[5] Scalable Video Compression, Vincent Botreau, [http://www.hd4u-itea.org/docs/Workshop%202/Scalable%20Video%20Compression\\_Vincent%20Botreau.pdf](http://www.hd4u-itea.org/docs/Workshop%202/Scalable%20Video%20Compression_Vincent%20Botreau.pdf)

[6] Richardson, E.G. Ian, 2003, H.264 and MPEG 4 Video Compression, WILEY

[7] Ridge, J.; Bao, Y.; Karczewicz, M.; Wang, X., 2005, "Fine-grained scalability for H.264/AVC," Signal Processing and Its Applications, 2005. Proceedings of the Eighth International Symposium on , vol.1, no., pp. 247-250.

[8] H. Schwarz, D. Marpe, T. Wiegand; Overview of the Scalable Extension of the H.264/AVC Video Coding Standard, JVT-W132 (final version), October 2006.

[9] H. Schwarz, D. Marpe, T. Wiegand; Basics Concepts for Supporting Spatial and SNR Scalability in the Scalable H.264/AVC Extension, IEEE IWSSIP 2005, September 2005, Greece.

[10] JSVM Software Manual, version: JSVM 6\_8\_2, October 2006.