

Hata Esnekliđi

Tahir Emre KALAYCI

Error Resilience: Hata Esnekliđi

İnternet Üzerinde Görüntü Akışı

- İnternet sadece “best-effort” servis sunar.
- İnternet Üzerinde Video Akış Problemleri:
 - Veriyolu genişliği (“Bandwidth”)
 - Gecikme “jitter” (“Delay Jitter”)
 - Kayıp oranları (“Loss Rates”)hakkında bir garanti yoktur.

Bu özellikler *Bilinmeyen* ve *dinamik* olarak bilinir. En önemli hedef internet üzerinden yüksek kalitede ve güvenilir video aktarımı için bir sistem tasarlamak.

Özet

- Hata mücadele yöntemleri 3 grupta incelenir:
 1. Kaynak ve kanal kodlayıcıda uygulana ve bitstream'i hatalara karşı esnek hale getiren yöntemler
 2. Çözücüde hataların tespit edilmesi sonucu ateşlenen ve oluşan hataları gizlemek için yöntemler
 3. Kaynak kodlayıcı ve çözücünün etkileşimini gerektiren ve kodlayıcının çözücüde yakalanan hatalara göre kendini uyumlaştırdığı yöntemler

1. Hata Esnek Kodlama

- “Robust Entropy Encoding”
 - Tekrar eşzamanlama işaretleri koyma
 - RVLC
 - Sözdizimine dayalı onarmalar
- Hata Esnek Tahminleme
 - Intra blok veya çerçevelerin eklenmesi
 - Bağımsız segment tahminleme

1. Hata Esnek Kodlama (Devam)

- Eşit olmayan hata korumalı katmanlı kodlama
- Çoklu tanımlama kodlama

2. Çözücü Hata Gizleme

- Doku bilgisinin düzeltilmesi (“recovery”)
 - “Motion Compensated Temporal Prediction”
 - Uzaysal Interpolasyon
 - Sonuç görüntünün düzgünlüğünün artırıldığı uzaysal ve zamansal interpolasyon
 - POCS Yöntemi
- Kodlama modları ve Hareket Vektörlerinin düzeltilmesi

3. Kodlayıcı ve Çözücü Etkileşimli

- Geribildirim bilgisine dayanan RPS (Referans Resim Seçimi)
- Geribildirim bilgisine dayanan Hata Takibi

Hata Esnekliđi

- *Bilinmeyen ve Dinamik problemler*
 - *Veriyolu Problemi → Oran Kontrolü*



1. Mevcut veriyolunu tahminleriz
 - Araştırma-Tabanlı (“Probe-Based”) veya Model-Tabanlı yöntemler
2. Video oranını mevcut veriyoluna göre ayarlarız
 - Gönderici veya alıcıda yerine getirilir

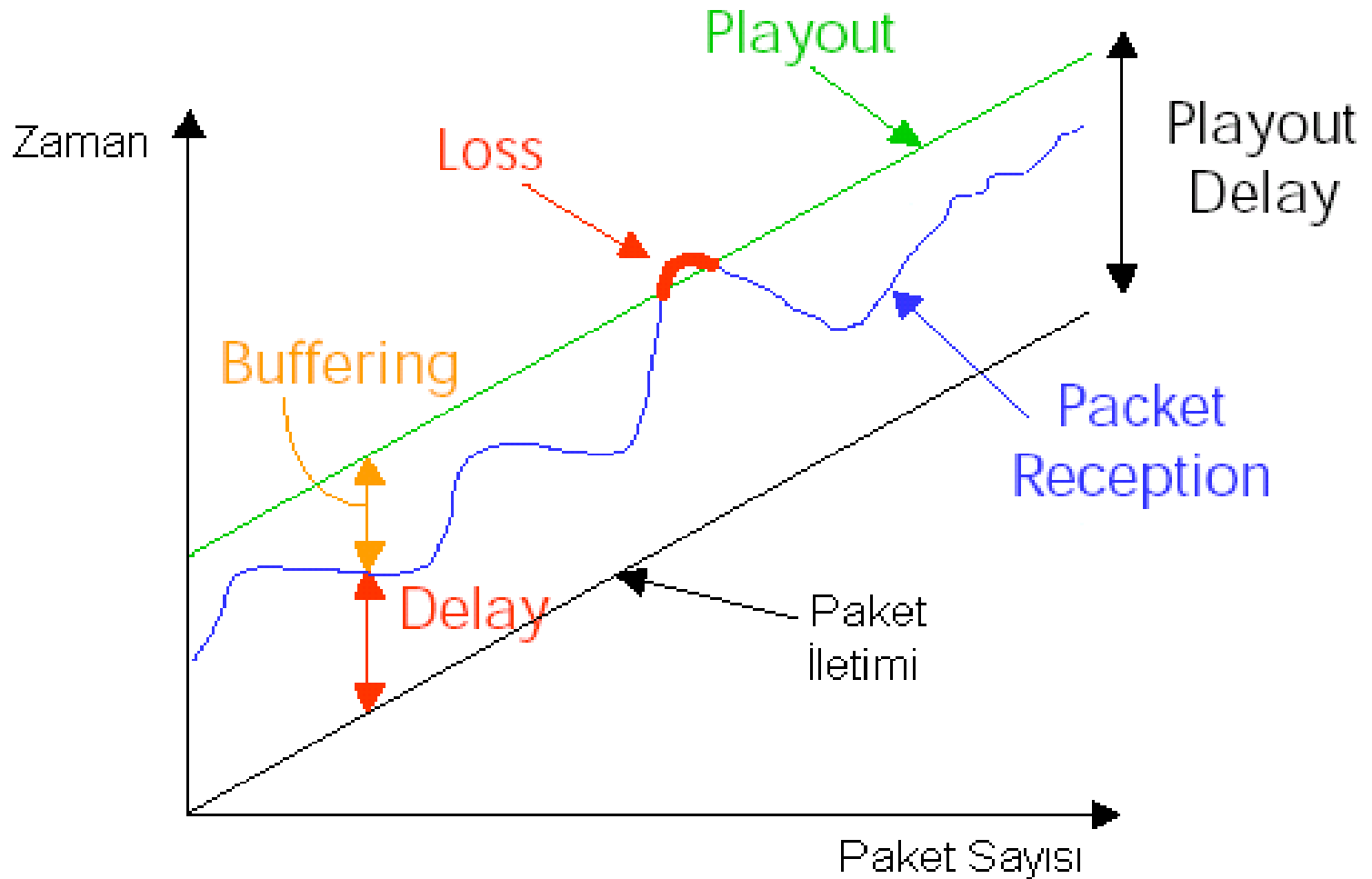
Hata Esnekliđi

- *Bilinmeyen ve Dinamik* problemler
 - Gecikme “jitter” → Oynatma (“playout”) tamponu



- Eğer görüntü verisi geç gelirse bu işe yaramaz
- Kısmi olarak bunun önüne geçmek için *oynatma tamponu* ekleyebiliriz.

Hata Esnekliği



Hata Esnekliği

- *Bilinmeyen ve Dinamik* problemler
 - *Kayıp* → *Hata Kontrolü*
 - İleri Hata Düzeltimi
(“Forward Error Correction”) (FEC)
 - Tekrar İletim (“Retransmission”)
 - **Hata Gizleme**
(“*Error concealment*”)
 - **Hata Esnek Kodlama**
(“*Error-resilient coding*”)

Hata Esnekliđi

Uygulama	Hata Karakteri
PSTN Üzerinde Görüntülü Telefon (“VideoPhone“)	Az bit hataları ve paket kayıpları
ISDN üzerinde görüntülü konferans	Pratik olarak hatasız (BER 10^{-8})
Sayısal Televizyon	Neredeyse hatasız
İnternet üzerinde görüntü akışlandırma	%0-30 paket kaybı
Kablosuz görüntü (Cep Telefonları)	10^{-3} ‘e kadar BER “burst” hatalar

Hata Kontrol

- Hata kontrolünün hedefi
 - Hataların etkilerini aşmaktır. Örneğin paket ağlarda paket kaybını, kablosuz ağlarda bit veya “burst” hatalarını aşmaktır.
 - Hata kontrolü tipleri:
 - İleri Hata Düzeltimi (“Forward Error Correction”) (FEC)
 - Tekrar İletim (“Retransmission”)
 - Hata Gizleme (“*Error concealment*”)
 - Hata Esnek Kodlama (“*Error-resilient coding*”)

Kanal
Kodlama

Kaynak
Kodlama

Hata Gizleme

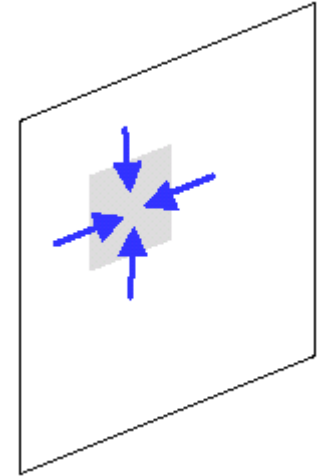
- Problem: İletim hataları **kayıp bilgi** sonucunu doğurabilir.
- Hedef: **Kayıp bilgiyi tahminleyerek** hata olduğu gerçeğini **gizlemek**.
- Hata gizleme decoder tarafında gerçekleştirilir.
- Görüntünün **uzaysal (“spatial”) ve zamansal (“temporal”) boyutlar boyunca kayda değer miktarda karşılıklı ilişkiyi** ortaya koyduğu gözlenlenmiştir.
- Yukarıdaki gözlemden yararlanılarak elimize doğru olarak erişmiş olan veri üzerinde uzaysal/zamansal interpolasyonlar uygulayarak kayıp veriyi tahminleriz.

Hata Gizleme

- Örnek olarak tek bir macroblock'u kaybettiğimizi düşünelim. Bu durumda aşağıdaki hata gizleme metodlarını uygulayabiliriz.

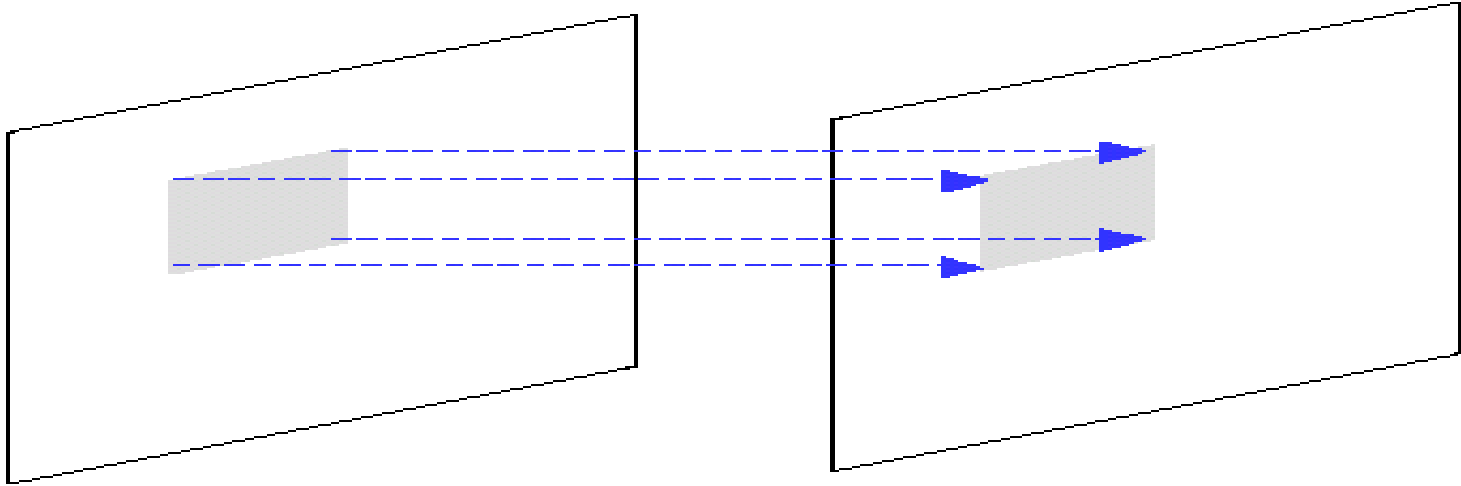
Hata Gizleme

- Uzaysal Interpolasyon
 - Kayıp pikselleri etraflarındaki piksellerden güzel bir şekilde (“smoothly”) anlam çıkararak (“extrapolating”) tahminleriz
 - Kayıp pikselleri tam olarak düzeltmek aşırı derecede zor bir iştir, bununla birlikte ortalama DC değerini doğru tahminlemek bile çok yardımcı olacaktır.



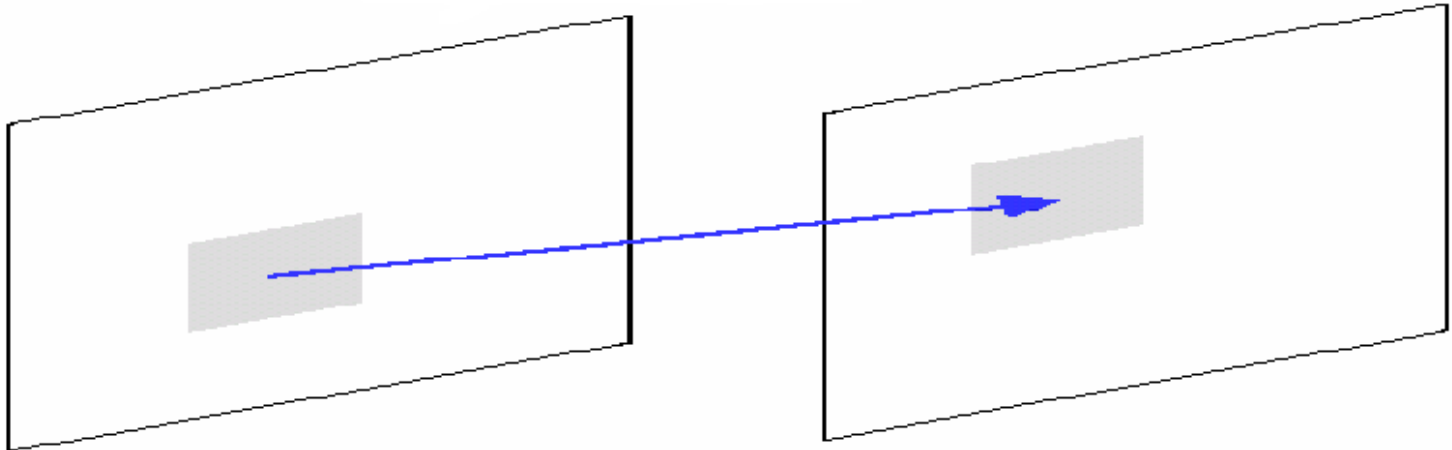
Hata Gizleme

- Zamansal (“Temporal”) Interpolasyon
(Dondurulmuş Çerçeve “Freeze Frame”)
 - Önceki çerçevedeki aynı uzaydaki pikselleri kopyala (“Freeze Frame”)
 - Hareket olmadığı durumlarda etkili, ama hareketin olduğu durumlarda problemlere sebep olabilir.



Hata Gizleme

- “Motion-compensated” Zamansal Interpolasyon
 - Önceki çerçevedeki “motion-compensated” blok olarak kullanarak elde edilen hareket vektöründen yararlanarak kayıp bloğu tahminleriz.
 - Kodlanmış hareket vektörü, komşusal hareket vektörü veya yeni hesaplanmış hareket vektörü kullanabiliriz.



Hata Gizleme

- Sonuçlar-Yorumlar
 - Genellikle tek bir macroblocktan fazlası kaybolur.
 - Uzaysal ve Zamansal interpolasyonların kombinasyonlarını kullanmalıyız.
 - Doğru hareket vektörü kullanıldığını varsayarsak motion-compensated zamansal interpolasyon en iyi gizlemeyi sağlar
 - Bir sürü ince (“sophisticated”) algoritma mevcuttur:
 - Bu problem sinyal kurtarma/tersleme (“recovery/inverse”) problemi olarak formüle edilebilir.
 - Hata gizleme decoder tarafında yapılmaktadır
 - Yeni hata gizleme algoritmaları, geleneksel decoderlere *standart uyumlu ilerlemeler* olarak eklenebilir.

Hata Esnekliğine Dayalı Görüntü Kodlama

- Hedef: Hatalara karşı esnek olabilecek kodlanmış “bitstream” ve görüntü sıkıştırma algoritması tasarlanmasıdır.
- Sıkıştırılmış görüntü hataya çok açıktır.
- Hata Örnekleri
 - VLC’de hata
 - Tahminlerde (“Prediction”) hata

Hata Esnekliğine Dayalı Görüntü Kodlama

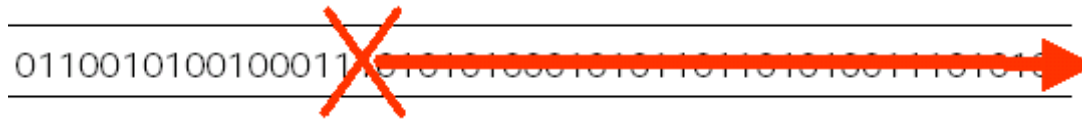
- Geleneksel MPEG vari bir sistem MC-Tahminleme, Blok-DCT, “runlength” ve Huffman kodlama içerir.
- Aşağıda anlatılan iki basit hata sınıfı ile karşılaşmamız olasıdır.

Hata Esnekliğine Dayalı Görüntü Kodlama

- “Bitstream” eşzamanlaması: Şifre çözücü hangi bitin hangi parametreyi temsil ettiğini bilmemektedir.
 - Örnek: Huffman codeword’ta veya başka bir değişken uzunlukta kodta (VLC) hata oluşması
 - Tek bitlik bir hata bile *devam eden kayda değer kayıplara* neden olur.
 - Not:
 - Sabit uzunluktaki kodlar (FLC) bu sorun kodların başlangıç ve sonlarının bilinmesi nedeniyle yoktur. Sadece tek bir codeword kaybıyla sınırlıdır. Bununla birlikte FLC’nin iyi bir sıkıştırma performansı sunmaması nedeniyle, VLC ve diğer entropy kodlama biçimleri tercih edilmektedir.

“Bitstream” Eşzamanlaması Problemini aşmak

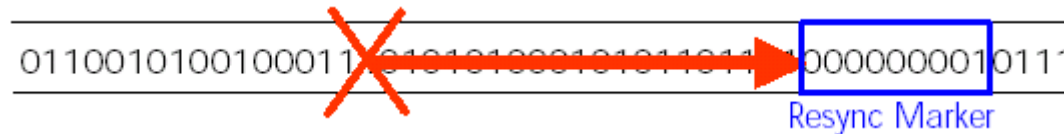
- Hedefimiz eşzamanlamayı tekrarlamaktır.



1. Çözüm:

– “Resync” İşareti Kullanımı

- a. Yapıcı tüm codewordlardan ve codeword birleşimlerinden vb. farklıdır.
- b. İşareten sonra şifre çözücüyü tekrar başlatmak için bilgi içerilir.



“Bitstream” Eşzamanlaması Problemini aşmak

- Eşzaman işaretleri *stratejik noktalara* koyulur. (Çerçeve, slice,... başlangıcı)

(MPEG-1/2, H.261/3)

- Her sabit sayıda blok, değişken sayıda bit tekrar eşzamanlanır.
- Etkin alanların bozulması daha muhtemeldir.
- Eşzaman işaretleri *periyodik* olarak koyulur. (MPEG-4)
 - Her sabit sayıda bit, değişken sayıda blok tekrar eşzamanlanır.
 - Avantajları:
 - » Etkin alanların bozulma ihtimali düşer.
 - » Yapıcıların aranması işlemi basitleşir.
 - » Ağ paketlemesini destekler.

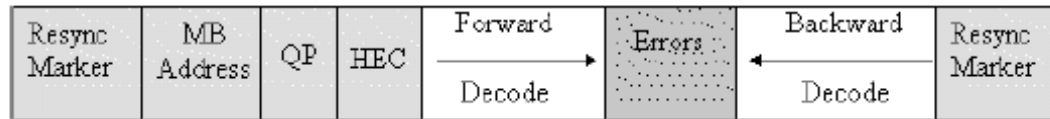
“Bitstream” Eşzamanlaması Problemini aşmak

2. Çözüm: *Tersinir Değişken Uzunlukta Kodlar* (“*Reversible Variable Length Codes - RLVC*”)

- Geleneksel VLC’ler sadece ileriye doğru çözülebilirler.
- RLVC’ler ayrıca *ters yönde* çözülme özelliğine sahiptir.
- Kullanımı: Bir hata yakalandığında sonraki tekrar eşzamanlamaya atlanır ve geriye doğru çözüme yapılmaya, böylece kısmi veri kurtarımı yapılır. (Veri kurtulamazsa atılır)



MPEG4
Sözdizimi
("Syntax")



“Bitstream” Eşzamanlaması Problemini aşmak

3. Çözüm

- *Veri Kısımlandırma (“Data Partitioning”)*
 - Gözlemler eşzaman işaretlerinden sonra gelen verilerin daha ilerideki verilerden daha doğru olduğunu ortaya çıkarmıştır.
 - Bu sonuca dayanarak “*En Önemli Bilgiler*” (MV’ler, şekil bilgisi, DC katsayıları) hemen eşzaman işaretlerinden sonraya koyulur. Ve daha az önemli olan bilgiler daha sonra koyulur. (AC katsayıları)



“Bitstream” Eşzamanlaması Problemini aşmak

- Buraya kadar konuştuğumuz çözümlerin sonucunda iki temel fikir oluşur:
 - Bozuk bilgi tecrit edilir(yerellenir)
 - Hızlı tekrar eşzamanlama sağlanır

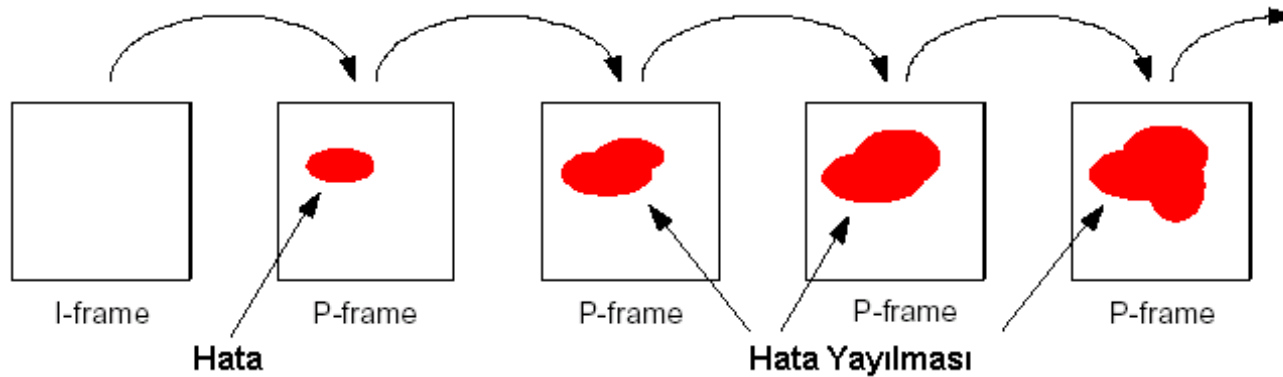
Uygulama-Farkında Paketleme

Uygulama Seviyesinde Çerçeveleme (ALF)

- Paket kaybıyla mücadele edebilmek için paket yükünün tasarımı kaybın etkisini azaltacak şekilde tasarlanır.
- Bu işi en iyi uygulama bilir. [Clark, Tennenhouse]
- Örneğin
 - Görüntü çözücü paket boyutunu bilebilir
 - Olası kayıp verinin sınırlarını bilebilir
 - Yük, her paket başında “bitstream” eşzamanlaması sağlanacak şekilde tasarlanır.
- RTP, MPEG-4 ve H.263+ aşağıdaki özellikleri destekler
 - İnternet üzerinde görüntü akışlandırması için “Bitstream” eşzamanlaması probleminin üstesinden gelir.
 - Hata Yayılımı temel engeldir.

Hatalı Durum ve Hata Yayılımı

- Bir hata çözücüdeki tekrar yapılmış çerçevenin (durum) hatalı olmasına sebep olur.
- Çözücünün durumu kodlayıcıdan farklı olur, bu da hatalı (yanlış eşlenmiş) tahminlemelere ve genellikle önemli hata yayılımlarına yol açar.



Hatalı Durum ve Hata Yayılımı Problemini aşmak

- Bir hata çözücünde tekrar yapılan çerçevenin yanlış olmasına, ve önemli hata yayılımına yol açar.
- Hedef: Hata Yayılımı etkisini azaltmak
(Örneğin tahminleme döngüsünü tekrar başlatmak)
- Çözümler:
 - Hepsi I-Frame:
 - Hata yayılımı engellenir ama sıkıştırma azdır.
 - Periyodik I-Frame: (Örnek MPEG GOP)
 - Örnek: Her 15 çerçevede bir I-Frame kullanımı
 - Hata yayılımını bir GOP'a kısıtlar
 - Sıkıştırma hala göreceli olarak azdır. (Çok düşük bit rate görüntüler için uygun değildir, örnek: kablosuz görüntü veya internet üzerinde görüntü)

Hatalı Durum ve Hata Yayılımı Problemini aşmak

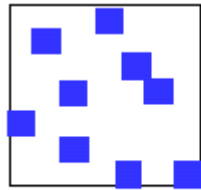
- Çözümler : (Devam)

- Her çerçevenin kısmi iç-kodlama (“intra-coding”)

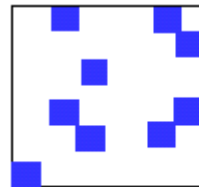
- Her “macroblock” iç-kodlanır.

- 2. Tüm MB’lerin Periyodik iç-kodlanması

- Her çerçevedeki MB’lerin belli bir yüzdesi önceden belirlenmiş bir sıralamayla iç-kodlanır; N çerçeveden sonra tüm MB’ler iç kodlanmış olur.



Frame n



Frame $n+1$

■ Intra-coded MB

- 3. “Preemptive” iç-kodlama hatalara açık olmanın tahminlenmesine dayanır. [Hinds, Pappas, Lim]

- Kanal kayıp modeline göre iç/gömülü (“intra”/“inter”) mod kararlarının optimize edilmesinde kullanılır.

Hatalı Durum ve Hata Yayılımı Problemini aşmak

- İç-kodlama yöntemlerinin avantajları
 - Hata yayılımının etkisini kısıtlar
 - Kodlayıcıda karmaşıklık (“sophistication”) vardır, çözücü basittir.
- Dezavantajlar
 - Yüksek bit rate gereklidir, kullanımı kısıtlanır
 - Optimal kullanım kanal karakteristiklerinin doğru bilinmesine dayanır.

Hata Yayılımı Problemini aşmak: Arka-Kanal İle Noktadan-Noktaya

- Özel Durum
 - Örneğin görüntülü telefon, görüntülü konferans



- Çözücü hatayı tespit eder ve kodlayıcıya bildirebilir
 - Tahminleme yeniden başlatılır (I-Frame gönderilir)
 - Basit, apaçık
 - Ancak iç kodlama için daha yüksek bit rate gerektirir
 - Sonraki tahminleme için hangi çerçeve referans olacak
 - Önceki temiz çerçevelerle gömülü kodlama kullanılır → daha iyi sıkıştırma
 - Kodlayıcı ve Çözücü birçok önceden kodlanmış çerçeve saklarlar.
 - Hangi önceden kodlanmış çerçevenin referans olarak kullanılacağını kodlayıcı seçer. (Örneğin sadece doğru alınmış çerçeveler kullanılır)
 - İşlemlerin iki modu vardır : ACK ve NACK

Hata Yayılımı Problemini aşmak: Arka-Kanal İle Noktadan-Noktaya

- Bu özel durumu özetlersek
 - Geri beslemenin olması önemli etkilere neden olur. (MPEG-4'te NewPRED, H.263'te Referans Resim Seçimi)
 - Ancak, kısa gidiş-dönüş gecikme(RTD)'ye sahip, güvenilir arka-kanal gerektirir.
 - RTD arttıkça, verimlilik azalır. (Çerçeve aralıkları olarak ölçülür.)
 - Çoğayayım, çokluyayım, arka-kanalı olmayan noktadan-noktaya ve önceden kodlanmış görüntü için uygun değildir.

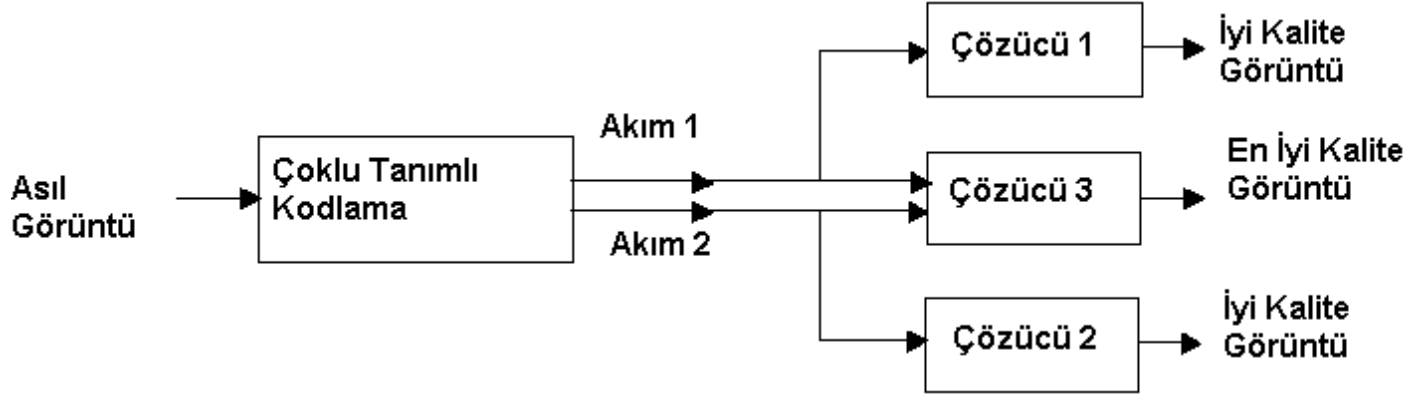
Ölçeklenebilir Görüntü Kodlama

- Görüntü bir temel katman ve tek yada daha fazla gelişme katmanına kodlanır.
- Örnekler: Zamansal, Uzaysal, SNR (Kalite) ölçeklenebilirliği
- Görüntü verisi öncelik sırasına göre değerlendirilir
- Farklı öncelikler güvenilir görüntü iletimini gerçekleştirmek için kullanılır. Örneğin eşit olmayan hata koruma, önceliklenmiş tekrara iletim
- Ölçeklenebilir görüntü kodlama farklı servis kaliteleri sunan ağlar için uygun bir eşlemedir.

Ölçeklenebilir Görüntü Kodlama

- Şu anki Internet “best-effort” tabanlıdır:
 - QoS desteklemez
 - Tüm paketler eşit derecede kaybolmaya yatkındır.
- Bundan başka temel katman kritiktir
 - Eğer temel katmanda bir hata oluşursa tüm görüntü kaybedilebilir.

Çoklu Tanımlı Görüntü Kodlama



- Görüntü birden fazla tanıma kodlanır, herbiri aşağı yukarı eşit öneme sahiptir.
- MD ("Multiple Description") kodlayıcının özellikleri
 - Herhangi bir akımı aldığımızda iyi kalitede görüntü
 - Her iki akımda aldığımızda en iyi kalitede görüntü elde edilir.

Çoklu Tanımlı Görüntü Kodlama

- MD Kodlama Yöntemleri
 - Tekrar eşzamanlama çerçeveleri ile birlikte çok sayıda iş parçacığı [Wenger]
 - Kestirimci MD “quantizer” [Vaishampayan, John]
 - MD Değişim (“Transform”) Kodlama [Reibman, Jafarkhani, Wang, Orchard, Puri]
 - Çoklu Durum [Apostolopoulos]

H.263'teki ER Araçları

- H.263 v3 8 aşağıdaki ER Araçları vardır:
 - BCH İleri Hata Düzeltme (FEC) (Annex H)
 - Esnek Eşzaman İşaretleyici Ekleme, “Slice Structured Mode” (Annex K)
 - Referans Resim Seçme (RPS- Annex N ve Annex U)
 - Ölçeklenebilirlik (Annex O)
 - Bağımsız Segment Kodlama (ISD- Annex R)
 - Veri Parçalama ve RVLC (Annex V)
 - Başlık Tekrarlama (Annex W)

MPEG-4'teki ER Araçları

- MPEG-4 Tekrar Eşzamanlama Araçları
- MPEG-4 Veri Bölümleme
- NEWPRED Modu
- RVLC ve Veri Bölümleme ile birlikte çalışan Veri kurtarma
 - Sözdizim Tabanlı Tamirler

JPEG2000 ER Araçları

- Hiyerarşik Tekrar Eşzamanlama ve Veri Bölümlenme
 - Bitstreamin altbandlar, bit-yüzeyleri ve bloklar şeklinde hiyerarşik organizasyonu
 - Tekrar eşzamanlama işaretçileri
 - Hata Esnek paket
- Hata Tespit ve Gizleme
 - Segmentasyon İşaretçileri
 - Bitstream sonu bilgisi
- QoS Kanalları kullanarak eşit olmayan hata koruma