

ÖZET

VEKTÖR NİCELEYİCİ

Vektör Niceleyici ve burada örneğin bir dönüşüm ses kodekinde etkin vektör nicelemesi için
usul. Usul, bir girdi hedef vektörün (s), çok sayıda sentroid ile karşılaştırılmasını içerir; her
5 sentroid, bir kod çizelgesindeki kod vektörlerinin ilgili bir sınıfını temsil eder. Ayrıca, kod
çizelgesinde girdi hedef vektör ile ilişkili bir arama için bir başlangıç noktası, karşılaştırmanın
sonucu bazında belirlenir. Kod çizelgesindeki kod vektörleri, her kod vektörü ve sınıfların
sentroidleri arasındaki mesafeyi yansıtan bir distorsiyon ölçüsüne göre sınıflandırılır. Vektör
Niceleyici ve usul, girdi vektör (s) ile ilişkili en olası aday kod vektörlerini içeren kod
10 vektörleri sınıfının, ilk aranabilmesini sağlar.

İSTEMLER

1. Bir Vektör Niceleyiciyi içeren bir dönüşüm kodaki (1100) tarafından yürütülen pik bölgesi kodlaması için usul olup, usul aşağıdakileri içerir:

aşağıdakilerin Vektör Niceleyici tarafından yürütülmesi için, bir bellekteki bilgisayar tarafından okunabilir komutları yürüten dönüşüm kodunun (1100) bir işlemcisinin (1206) çalıştırılması:

- bir girdi hedef vektörün (s), dört sentroid ($C_0, C_1, C_{0,döndürülmüş}$ ve $C_{1,döndürülmüş}$) ile karşılaştırılması (902); burada $C_{0,döndürülmüş}$ sentroidi, C_0 sentroidinin bir döndürülmüş versiyonudur ve $C_{1,döndürülmüş}$ sentroidi, C_1 sentroidinin bir döndürülmüş versiyonudur; burada $C_{i,döndürülmüş}$ 'ün elemanları, $C_{i,döndürülmüş} = [C_i(M) C_i(M-1) \dots C_i(1)]$ olarak tanımlanmıştır; burada $C_i(m)$, C_i sentroidinin vektör elemanlarıdır; M , vektörün bir uzunluğudur ve i , sentroidin bir indeksidir; her sentroid, kod vektörlerinin ilgili bir sınıfını temsil eder ve bir kümeleme algoritmasının yürütülmesinin bir sonucu olarak elde edilir;

- kod çizelgesinde girdi hedef vektör ile ilişkili bir arama için bir başlangıç noktasının, karşılaştırmanın sonucu bazında belirlenmesi (906) ve

- kod çizelgesinde, belirlenmiş başlangıç noktasında başlayan ve girdi hedef vektörü (s) temsil etmek üzere bir kod vektörünü tanımlayan bir arama

gerçekleştirilmesi (910); burada blok veya zaman dilimi başına girdi hedef

vektörlerin bir sayısı değişkendir ve bir arama alanı, arama alanının boyutu girdi hedef vektörlerin sayısına bağlı olacak şekilde dinamik olarak ayarlanır; söz

konusu girdi hedef vektörler, bir spektral pik bölgesini temsil eder; burada kod çizelgesi, kod çizelgesindeki kod vektörlerinin, bir distorsiyon ölçüsüne göre

sınıflandırılacağı şekilde oluşturulur; distorsiyon ölçüsü, her kod vektörü ve söz konusu C_0 ve C_1 sentroidleri arasındaki mesafeyi yansıtır.

2. İstem 1'e göre usul olup, burada arama alanı, arama alanının boyutunun hedef vektörlerin sayısı bazında arttırılması veya azaltılması vasıtasıyla dinamik olarak ayarlanır.

3. İstem 1 veya 2'ye göre usul olup, burada kod çizelgesindeki kod vektörleri, bir birinci sentroide (C_0) en yakın ve bir ikinci sentroide (C_1) en uzak mesafedeki kod kelimelerinin, kod çizelgesinin bir sınıfında olacağı, C_1 'e en yakın ve C_0 'a en uzak kod kelimelerinin ise kod çizelgesinin başka bir sınıfında kümeleneceği şekilde

sınıflandırılır.

4. Pik bölgesi kodlaması yürütmek üzere uyarlanmış, bir Vektör Niceleyiciyi (1000) içeren dönüşüm kodeki olup, aşağıdakileri içerir:

5 _ blok veya zaman dilimi başına girdi hedef vektörlerin (s) bir değişken sayısını elde etmek üzere uyarlanmış iletişim birimi (1002);

 - bir girdi hedef vektörü (s), dört sentroid (C_0 , C_1 , $C_{0,döndürülmüş}$ ve $C_{1,döndürülmüş}$) ile karşılaştırmak üzere uyarlanmış bir karşılaştırma birimi (1004); burada $C_{0,döndürülmüş}$ sentroidi, C_0 sentroidinin bir döndürülmüş versiyonudur ve $C_{1,döndürülmüş}$ sentroidi, C_1 sentroidinin bir döndürülmüş versiyonudur; burada

10 $C_{i,döndürülmüş}$ 'ün elemanları, $C_{i,döndürülmüş} = [C_i(M) C_i(M-1) \dots C_i(1)]$ olarak tanımlanmıştır; burada $C_i(m)$, C_i sentroidinin vektör elemanlarıdır; M , vektörün bir uzunluğudur ve i , sentroidin bir indeksidir; her sentroid, kod vektörlerinin ilgili bir sınıfını temsil eder ve bir kümeleme algoritmasının yürütülmesinin bir sonucu olarak elde edilir;

15 - kod çizelgesinde bir arama için bir başlangıç noktasını, karşılaştırmanın sonucu bazında belirlemek ve bir arama alanını, arama alanının boyutu, girdi hedef vektörlerin sayısına bağlı olacak şekilde dinamik olarak ayarlamak üzere uyarlanmış bir belirleme birimi (1008); söz konusu girdi hedef vektörler, bir spektral pik bölgesini temsil eder ve

20 - kod çizelgesinde, belirlenmiş başlangıç noktasında başlayan ve girdi hedef vektörü (s) temsil etmek üzere bir kod vektörünü tanımlayan bir arama gerçekleştirmek üzere uyarlanmış bir arama birimi (1010);

 burada kod çizelgesindeki (1014) kod vektörleri, her kod vektörü ve söz konusu C_0 ve C_1 sentroidleri arasındaki mesafeyi yansıtan bir distorsiyon ölçüsüne göre

25 sınıflandırılır.

5. İstem 4'e göre dönüşüm kodeki olup, burada belirleme birimi (1008), arama alanının boyutunun hedef vektörlerin sayısı bazında arttırılması veya azaltılması vasıtasıyla arama alanını dinamik olarak ayarlamak üzere uyarlanmıştır.

6. İstem 4 veya 5'e göre dönüşüm kodeki olup, burada kod çizelgesindeki kod vektörleri, bir birinci sentroide (C_0) en yakın ve bir ikinci sentroide (C_1) en uzak mesafedeki kod kelimelerinin, kod çizelgesinin bir sınıfında olacağı, C_1 'e en yakın ve C_0 'a en uzak kod kelimelerinin ise kod çizelgesinin başka bir sınıfında kümeleneceği şekilde sınıflandırılır.
- 30

7. Mobil terminal olup, İstem 4-6'dan herhangi birine göre dönüşüm kodekini içerir.
8. Bilgisayar programı (1210) olup, bir Vektör Niceleyiciyi içeren bir dönüşüm kodekinin bir işlemcisinde yürütüldüğünde, Vektör Niceleyicinin, İstem 1-3'ten herhangi birine göre karşılık gelen pik bölgesi kodlaması usulünü yürütmesine neden olan bilgisayar tarafından okunabilir kodu içerir.
9. Bilgisayar program ürünü (1208) olup, bir bilgisayar tarafından okunabilir ortamı ve bilgisayar tarafından okunabilir ortamda kayıtlı İstem 8'e göre bir bilgisayar programını (1210) içerir.

TARİFNAME

VEKTÖR NİCELEYİCİ

Buluşun Açıklaması

Buluşun Teknik Alanı

- 5 Önerilen teknoloji, genel olarak vektör nicelemeye (VQ) ve özellikle bunun hassasiyetine ve bilgi işlem karmaşıklığına ilişkindir.

Buluşla İlgili Bilinen Hususlar

- Niceleme algoritmalarının iki ana sınıfı, yani bir girdi vektörün elemanlarını tek tek işleyen skaler niceleyiciler (SQ'ler) ve bir girdi vektörü tek bir birim olarak işleyen (tüm vektör
- 10 boyutları birlikte nicelenir) vektör niceleyiciler (VQ'ler) mevcuttur. Belirli bir bit hızında VQ'ler, SQ'lerden üstündür, ama bu, artmış bilgi işlem karmaşıklığı ve bellek depolama alanı pahasına gerçekleşir.

Nicelenecek hedef vektör M boyutlu olsun: $\mathbf{s} = [s(1) \ s(2) \ \dots \ s(M)]$. VQ algoritması, önceden kaydedilmiş M boyutlu kod vektörlerinin ($\mathbf{c}_k = [c_k(1) \ c_k(2) \ \dots \ c_k(M)]$) K boyutlu bir kod

- 15 çizelgesinde (CB) $\{\mathbf{c}_k\}_{k=1}^K$ bir arama gerçekleştirir. Böyle bir arama, bir distorsiyon ölçüsü ($d(\mathbf{s}, \mathbf{c}_k)$) bazında en iyi eşleşmeyi (k^{opt}) sağlayan kod çizelgesi vektörünün indeksini verir. Aşağıdaki denklemler (1-2), arama ölçütünün bir karesel hataya dayandığını varsayarak bu işlemi açıklamaktadır:

$$k^{opt} = \arg \min_k d(\mathbf{s}, \mathbf{c}_k) \quad (1)$$

$$d(\mathbf{s}, \mathbf{c}_k) = \sum_{m=1}^M (s(m) - c_k(m))^2 \quad (2)$$

20

Optimal indeks (k^{opt}) kod çözücüye transfer edilir ve karşılık gelen kod vektörü CB'den çıkartılır (eş CB'ler, hem kodlayıcıda hem kod çözücüde mevcuttur) ve hedef vektörü yeniden oluşturmak için kullanılır. Tipik olarak CB, çevrim dışı çalıştırılır ve verilerin istatistiksel özelliklerini yakalar. Birçok durumda, basit karesel hata (bkz. denklem (2)) ağırlıklar ile

25 modifiye edilir, böylece:

$$d(\mathbf{s}, \mathbf{c}_k) = \sum_{m=1}^M w(m) \cdot (s(m) - c_k(m))^2 \quad (3)$$

burada ağırlıklar, $(w(m))$ uygulamaya bağlıdır. Buradaki sunumun basitliği için bundan sonraki açıklamada yalnızca denklem (2)'de tanımlanan karesel hata kullanılacaktır. Ancak burada açıklanan prensiplerin, denklem (3)'teki gibi daha sofistike ölçütler kullanıldığında da geçerli olduğu dikkate alınmalıdır..

- 5 Yukarıdaki açıklamadan anlaşılaçağı gibi, yeniden oluşturulmuş hedef sinyalin hassasiyeti veya kalitesi, kod çizelgesinin boyutuna (K) bağlıdır; burada büyük bir CB, küçük bir CB'den daha yüksek hassasiyete ve dolayısıyla daha iyi kaliteye yol açar. Aynı zamanda, denklem (1)'den, ana bilgi işlem karmaşıklığının da CB'nin boyutuyla ilişkili olduğı sonucu çıkarılabilir; burada vektör boyutluluğunun uygulama tarafından belirlendiğı varsayılmıştır.
- 10 Tipik olarak, ses iletim sistemleri, sınırlandırılmış bilgi işlem karmaşıklığı kısıtlamaları altında oluşturulur. Yani en kötü durum karmaşıklığı, belirli bir önceden tanımlanmış seviyeyi (L_{MAX}) aşmamalıdır. Örneğın bir ses kodekinin bilgi işlem karmaşıklığı, tipik olarak Saniyede Ağırlıklandırılmış Milyon İşlem (WMOPS) ile ölçölür, ama bir VQ modölünü dikkate aldığımızdan, karmaşıklık, doğrudan arama alanının boyutu (CB'nin boyutu) ile ilişkilidir. Bir
- 15 VQ, tipik olarak bir kodekteki en karmaşık modöldür ve ayrıca, VQ aramasını bu kadar karmaşık kılan, CB aramasıdır (CB vektörleri ile karşılaştırmaların sayısı).

Eğer bir VQ sistemi, bir defada bir hedef vektörü (S) niceleyecekse, arama alanının (K), karmaşıklığın L_{MAX} 'ı aşmayacağı şekilde optimize edilmesi gereklidir. Bölünmüş ve çok kademeli VQ gibi bazı çevrim dışı optimizasyon teknikleri, vektörün (s) özellikleri ve yeniden

- 20 oluşturulmuş vektörün kalite koşulları dikkate alındığında, karmaşıklıkta (ve depolama alanında) belirli bir azalma sağlayabilir.

Eğer VQ sistemi, değişen sayıda (N) vektörle bir defada birden fazla hedef (girdi) vektörü $\{s_n\}_{n=1}^N$ niceleyecekse, yukarıda belirtilen çevrim dışı optimizasyon teknikleri, karmaşıklık ve kalite kısıtlamalarını karşılayamamaktadır. Böyle durumlarda, çevrim dışı optimizasyon,

- 25 A) çok sayıda girdi vektör eş zamanlı olarak niceleneceğı zaman, karmaşıklığın sınırlandırılmasını (=aramanın sınırlandırılması) ve B) az sayıda vektör niceleneceğı zaman, yüksek hassasiyetin korunmasını (=büyük bir kod çizelgesinde arama) içeren çelişen koşullar arasında bir denge bulmak zorundadır; bu, basit bir iş değildir.

Bilinen teknik belgesi "Fast codebook search algorithm for unconstrained vector quantization", Chen C-Q ve diğerleri, her girdi vektör için T küme arasından C aday kümenin seçildiğı bir vektör niceleme usulünü açıklamaktadır. Tam arama, yalnızca bu C aday kümede yürütölür.

- 30

Diğer yandan bilinen teknik belgesi "Application of sorted codebook vector quantization to spectral coding of speech", Mohammadi H.R.S. ve diğerleri, kod çizelgesinin bir ön işlem aşamasında önceden tanılanmış ölçütlere göre sınıflandırıldığı bir vektör niceleme usulünü açıklamaktadır. Kodlama aşamasında, ilk olarak girdi vektörün pozisyonu hesaplanır ve ardından çevre bölgede ofsetle birlikte tam arama yürütülür.

Buluşun Özet Açıklaması

Burada açıklanan teknoloji, örneğin girdi akımında kayıplı sıkıştırma gerçekleştiren ses ve video sıkıştırma / iletim sistemlerine uygulanabilir ve birkaç farkı yönde açıklanabilir. Burada açıklanan teknoloji, sınıflara bölünmüş ve sıralanmış bir kod çizelgesini ve nicelenecek bir girdi hedef vektörün (s), söz konusu sınıflardan birinde sınıflandırılmasını içerir. Burada açıklanan teknoloji, girdi vektör (s) ile ilişkili aday kod vektörlerinin en olası kümesini içeren kod çizelgesindeki kod vektörü sınıfının, kod çizelgesindeki sınıflar arasında ilk aranmasını sağlar. Dolayısıyla, girdi vektör (s) için en iyi eşleşen kod vektörü, bir aramada daha erken bulunabilir ve bilgi işlem karmaşıklığı azaltılabilir.

Buluş, bağımsız istemlerde tanımlandığı gibidir. Tercih edilen düzenlemeler, bağımlı istemlerde belirtilmiştir.

Kod çizelgesinde bir arama gerçekleştirilen bir arama bölgesinin boyutu, girdi hedef vektörlerin bir sayısı ve bir maksimum karmaşıklık kısıtlaması bazında uyarlanabilir.

Kodlama birimi başına girdi hedef vektörlerin sayısı değişken olabilir. Ayrıca maksimum karmaşıklık kısıtlaması, dinamik olarak ayarlanabilir. Ayrıca, kod çizelgesinde belirlenmiş arama alanında, belirlenmiş başlangıç noktasından başlayarak bir arama gerçekleştirilebilir; burada arama, girdi hedef vektör (s) için en iyi eşleşmeyi verir. Burada "en iyi eşleşme" ile girdi hedef vektör ve kod çizelgesindeki bir adaş vektör arasındaki bir mesafe ölçüsüyle bağlantılı olarak en yakın eşleşme, en kısa mesafe kastedilmektedir, yani en iyi eşleşme, mesafe ölçüsüne göre girdi hedef vektöre en kısa mesafeye sahip bir kod vektörüdür.

Şekillere Yönelik Özet Açıklama

Şimdi, önerilen teknoloji, örnek düzenlemeler vasıtasıyla ve ilişikteki çizimlere referansla daha detaylı açıklanacaktır; bu çizimlerde:

Şekil 1a ve 1b, burada açıklanan bir çözüme göre bir sıralanmış CB'nin yapısını göstermektedir. Arama, nokta 0'dan veya nokta 1'den başlayıp CB'nin diğer ucuna doğru gider.

Şekil 2, simetriden yararlanan örnek bir CB yapısını göstermektedir. Bir ROM belleğinde yalnızca C_0 ve C_1 kod vektörleri kayıtlıdır.

Şekil 3, girdi vektörün (s), her biri bir kod çizelgesindeki bir sınıf ile ilişkilendirilmiş birkaç sentroidin ($\{ C_0 C_1 C_{1,döndürülmüş} C_{0,döndürülmüş} \}$) her biriyle karşılaştırılması vasıtasıyla, bir girdi vektör (s) için bir optimal sınıfı belirlemeye yönelik örnek bir fonksiyonel birimi betimlemektedir.

Şekil 4, bir kod çizelgesindeki bir arama bölgesinin bir boyutunun, mevcut bir çerçevede tespit edilmiş spektral piklerin bir sayısı ve muhtemelen kodun bit hızı bazında belirlenmesi için bir fonksiyonel birimi betimlemektedir.

Şekil 5, çerçeve başına piklerin sayısı azaldıkça bir arama bölgesinin arttığını betimleyen bir tablodur. Örnekte, 17 pik (=17 girdi vektör) ile arama, yalnızca (bu örnekte minimum arama alanı olarak tanımlanan) 7 bitlik CB'de gerçekleştirilir, ama 8 veya daha az pik ile arama, 8 bitlik CB'de (maksimum arama alanı) gerçekleştirilir, çünkü bu, maksimum karmaşıklık kısıtlaması altında "karşılanabilir".

Şekil 6a-d, farklı arama bölgelerinin örneklerini göstermektedir.

Şekil 7, izin verilen karmaşıklığın (L_{MAX}), sisteme bir dış antiteden bildirilebileceğini betimlemektedir. L_{MAX} parametresi, örneğin CPU yüküne veya batarya durumuna dayanabilir.

Şekil 8, önerilen teknolojiye kullanılacak bir kod çizelgesinin (CB) oluşturulması için bir prosedürdeki eylemleri betimleyen bir akış şemasıdır.

Şekil 9a-c, burada önerilen teknolojinin örneklerine göre vektör nicelemeye (VQ) yönelik prosedürlerdeki eylemleri betimleyen akış şemalarıdır.

Şekil 10, burada önerilen teknolojinin bir örneğine göre bir vektör niceleyiciyi betimleyen bir blok diyagramıdır.

Şekil 11, burada önerilen teknolojinin bir örneğine göre bir vektör niceleyiciyi içeren bir kodun betimleyen bir blok diyagramıdır.

Şekil 12, burada önerilen teknolojinin bir örneğine göre vektör niceleme için bir düzeneği betimleyen bir blok diyagramıdır.

Buluşun Detaylı Açıklaması

Kısaca açıklandığı gibi, burada tarif edilen çözüm, (blok veya zaman aralığı başına) herhangi bir sayıda hedef (girdi) vektör için belirli bir karmaşıklık kısıtlaması çerçevesinde, yüksek hassasiyette ve dolayısıyla kalitede bir niceleme elde edilecek şekilde bir VQ'nün arama alanının dinamik olarak uyarlanmasına ilişkindir. Yani bilgi işlem karmaşıklığı koşulları (bkz. L_{max}) ihlal edilmeyecektir. Bu, aramanın özel bir sınıflandırılmış ve sıralanmış CB'de gerçekleştirilmesi vasıtasıyla sağlanır. Her hedef vektör için arama alanında başlangıç noktası, bir sınıflandırma prosedürüne dayanır ve arama alanının boyutu, hedef vektörlerin sayısı bazında artırılır veya azaltılır. Burada açıklanan VQ algoritması, verilerin ne olduğundan bağımsız olarak veri sıkıştırma için bir "araç" olarak değerlendirilebilir, yani veriler örneğin video ve/veya ses olabilir. Bu açıklamada VQ, bir ses kodeki bağlamında açıklanmıştır, ama burada açıklanan konsept, ses kodekleri ile sınırlı değildir. Örneğin video kodeklerinde de uygulanabilir.

Burada açıklanan algoritma, özel olarak dizayn edilmiş bir CB'ye dayanmaktadır. Böyle bir kod çizelgesinin bazı varyantları, aşağıda daha detaylı açıklanacaktır. İlk olarak, temel bir durum açıklanacak ve ayrıca aşağıda daha gelişmiş bir şema tartışılacaktır. CB'nin kod vektörleri, bir çevrim dışı kipte burada açıklanan çözüme göre düzenlenebilir.

Özel olarak dizayn edilmiş avantajlı CB'nin temel bir versiyonuna göre, bir CB'nin kod vektörleri, burada C_0 ve C_1 olarak belirtilen iki sınıfa bölünmüştür (bu notasyon, hem sınıfların adları için hem karşılık gelen sentroidler için kullanılacaktır; bkz. Şekil 1). Verileri iki sınıfa ayırmak için K-ortalamlar algoritması (Genelleştirilmiş Lloyd algoritması) kullanılabilir. Tüm bir veri kümesini ve istenen sayıda sınıfı girdi olarak alan ve istenen sayıda sınıfın sentroidlerini çıktı olarak veren iyi bilinen bir tekniktir. Örneğin eğer istenen sayıda sınıf 2 olarak belirtilmişse algoritma, çıktı olarak 2 sentroid vektörü verir. Bu sentroidlerin, bir K-ortalamlar algoritması kullanıldığında veri kümesinin vektörleri ile aynı boyutta vektörler oldukları, ama veri kümesine ait olmadıkları dikkate alınmalıdır. Yani sentroid vektörleri, CB'nin dışındadır ve mevcut bazı kod vektörleri ile çakışmaları gerekli değildir. Burada "sentroid" ile genel olarak vektörlerin bir sınıfını temsil eden bir referans vektör kastedilmektedir.

Bu durumda CB'deki tüm kod vektörleri, örneğin denklem (4)'te tanımlanan gibi bir distorsiyon ölçüsüne göre sınıflandırılır

$$d(c_i, C_0, C_1) = \sum_{m=1}^M (c_i(m) - C_0(m))^2 + \sum_{m=1}^M (c_i(m) - C_1(m))^2 \quad (4)$$

Yukarıdaki distorsiyon ölçüsü, C_0 'a yakın kod vektörler için büyük negatif değerler ve C_1 'e yakın kod vektörleri için büyük pozitif değerler ile sonuçlanır veya bunu varsayar. İki sınıfın sentroidlerinden (C_0 ve C_1) eşit mesafede bulunan kod vektörleri, sıfıra yakın bir distorsiyon ölçüsü üretir. CB'de kod vektörleri, Şekil 1a ve b'de gösterildiği gibi örneğin distorsiyon ölçüsünün artırılması vasıtasıyla sıralanır.

Her girdi hedef vektör, iki sentroid (iki sınıfın ilgili sentroidi) ile karşılaştırılır ve sonucu bağlı olarak ya C_0 sınıfına ya da C_1 sınıfına atanır, yani bunlardan birine ait olduğu sonucu çıkarılır veya belirlenir. Bu sınıflandırma bazında, aramanın başlangıç noktası, ya (hedef vektör C_0 sınıfına ait olduğunda) en üst nokta (Şekil 1a) ya en sol (Şekil 1b) nokta 0 olacak ya da (hedef vektör C_1 sınıfına ait olduğunda) en alt nokta (Şekil 1a) veya en sağ (Şekil 1b) nokta 1 olacak şekilde seçilir. Şimdi, arama alanının boyutu, blok veya zaman dilimi / aralığı başına girdi hedef vektörlerin sayısına (N) bağlı kılınmalıdır. Arama alanı (K) tüm CB'nin boyutu olmayacak, ama değişken olacak şekilde yeniden tanımlanırsa, burada açıklanan uyarılamanın gerisindeki konsept, denklem (5)'te tanımlanabilir

$$N \times K \approx \text{sabite} \quad (5)$$

Başka bir deyişle $\#Niceleyiciler \times \#Niceleyici \text{ başına işlem} \approx \text{sabite}$; burada "Niceleyici", bir girdi vektörü kod vektörlerinden birine eşleyen algoritma olarak değerlendirilebilir.

Burada, bir örnek olarak VQ, spektral pikleri ve kesin olarak spektral piklerin etrafındaki bölgeleri kodlayan bir dönüşüm kodeki bağlamında açıklanmıştır. Böyle bir kodek bağlamında, bir girdi hedef vektör, işlenmekte olan ses sinyalinin bir bölümünün bir spektral pikini (bölgesini) yansıtabilir. Bir ses sinyalinin bir zaman diliminin, örneğin 30 ms'nin sinyal spektrumundaki spektral piklerin sayısı, bu zaman diliminde ses sinyalinin spektral özelliklerine bağlıdır. Bir ses sinyalinin spektral özellikleri, zamanla değişebildiğinden ve örneğin farklı tiplerde ses için farklı olduğundan, spektral piklerin sayısı, farklı zaman dilimleri arasında ve farklı ses sinyalleri arasında değişebilir. Dolayısıyla spektral pik bölgelerini kodlayan bir dönüşüm kodlayıcısı durumunda, blok veya zaman dilimi başına VQ için girdi vektörlerin sayısı değişecektir. Buradaki örneklerde, bir ses sinyalinin bir zaman diliminde spektral piklerin bir sayısına karşılık gelen girdi vektörlerin maksimum sayısı 17'dir. Ancak bu sayı, yalnızca bir örnektir ve genel olarak çözümü sınırlayıcı olarak yorumlanmamalıdır.

Aslında, yukarıda açıklanan şema, VQ için gerekli işlemlerin sayısını dar (veya neredeyse sabit) bir aralıkta tutar; yani VQ'lerin sayısı arttığında, yani girdi hedef vektörlerin sayısı arttığında, VQ başına işlemlerin sayısı azalır (arama alanının boyutu azalır / CB'nin yalnızca bir kısmı aranır), böylece karmaşıklık koşulları (yani kısıtlamaları) ihlal edilmez. N 'nin azalmasıyla birlikte arama alanı (K), en çok tüm CB'nin boyutuna kadar arttırılabilir; bu, yeniden oluşturulmuş vektör için daha yüksek hassasiyete ve dolayısıyla kaliteye yol açar. Bir vektör niceleyicinin hassasiyeti, bir orijinal sinyal ve karşılık gelen yeniden oluşturulmuş veriler arasındaki bir karesel hata olarak ölçülebilir.

Bu şekilde, VQ'nün kod çizelgesinin en kötü durum senaryosu (yani girdi hedef vektörlerin maksimum sayısı) için dizayn edilmesine gerek yoktur. Aksine, örneğin en iyi durum senaryosu için dizayn edilebilir, dolayısıyla muhtemelen girdi hedef vektörlerin maksimum sayısı için maksimum karmaşıklık kısıtlaması (L_{MAX}) çerçevesinde aranabilecek olandan fazla kod vektörünü içerir. Maksimum karmaşıklık koşulu, aramanın kapsamı vasıtasıyla karşılanacaktır, yani CB'deki arama alanının boyutu, girdi hedef vektörlerin sayısına bağlıdır. Ancak, bu "körlemesine", örneğin burada önerilen CB olmaksızın yapılırsa, nicelemenin kalitesi büyük ölçüde zarar görecektir, çünkü en iyi eşleşme vektörünün CB'de nerede bulunduğunu veya bu en iyi eşleşme vektörünün, arama alanı küçültüldüğünde kod çizelgesinin aranacak bir bölümde bulunup bulunmadığını bilmenin bir yolu olmayacaktır. Bu sorun, burada açıklanan kod çizelgesinin özel dizaynı vasıtasıyla çözülmüştür. Burada açıklanan CB dizaynının, kodlama birimi başına girdi vektörlerin sayısının sabit olduğu uygulamalar için de faydalı olduğu dikkate alınmalıdır.

Örnek düzenleme: Spektral pik bölgelerinde kısıtlanmış VQ

Hedef vektörlerin (s) bir kümesi, bir dönüşüm alanı ses kodlamasında spektral pik bölgelerini, örneğin MDCT piklerinin çevre bölgesinde dönüşüm katsayılarını temsil eder. Dolayısıyla bu bağlamda, hedef vektörlerin sayısı zamanla değişir, çünkü spektral piklerin sayısı, bir zaman blokundan diğerine değişir.

Bu tip uygulamada (pik bölgesi kodlaması), hedef vektörler (S), CB'yi daha fazla optimize etmek için kullanılabilen belirli simetritler sergiler. Örneğin bir spektral pikin her iki tarafındaki dönüşüm katsayıları, benzer istatistiklere sahiptir. Hedef vektörlerin (S) pik pozisyonunda merkezlendiğini varsayarsak, yukarıda açıklanan simetri, sıralanmış CB'ye Şekil 1a ve 1b'den ek yapı ilave edilmesine olanak sağlar. Böyle yeni, ek bir geliştirilmiş CB'nin yapısı Şekil 2'de betimlenmiştir. Şekil 2, sol bölümün kod vektörlerinin bir Salt Okunur Bellek gibi bir belleğe kaydedildiği bir CB'yi betimlemektedir. Ancak CB'nin sağ tarafı için,

yani $C_{1,döndürülmüş}$ ve $C_{0,döndürülmüş}$ sınıfları için önceden kaydedilmiş kod vektörleri yoktur. Bu sınıfların kod vektörleri, CB'nin sağ tarafının kod vektörlerinin döndürülmüş versiyonlarıdır. Dolayısıyla $C_{0,döndürülmüş}$ ve $C_{1,döndürülmüş}$ sınıflarında bir arama gerçekleştirirken, arama, sol tarafın belleğe kaydedilmiş kod vektörlerinde, ama kod vektörlerinin elemanları merkezin

5 etrafında döndürülmüş durumda gerçekleştirilir, böylece kod vektörleri ($\mathbf{c}_{k,döndürülmüş}$) denklem (6) ile elde edilir

$$\mathbf{c}_{k,döndürülmüş} = [c_k(M) \quad c_k(M-1) \quad \dots \quad c_k(1)], \quad (6)$$

burada $c_k(m)$, kayıtlı CB'de karşılık gelen sınıfın (C_j) (yani C_0 veya C_1) vektör elemanlarıdır. Yani C_0 'da belirli bir kod vektörünün elemanları $\{C_{01} C_{02} C_{03} C_{04}\}$ ise, $C_{0,döndürülmüş}$ 'te karşılık

10 gelen bir kod vektörünün elemanları $\{C_{04} C_{03} C_{02} C_{01}\}$ 'dir.

Şekil 2'de betimlenen gibi bir CB kullanılırken, bir girdi hedef vektör, dört sentroid ile karşılaştırılır ve arama için bir başlangıç noktası belirlemek amacıyla bir sınıfa atanır, yani girdi hedef vektör için optimal sınıf, girdi vektörün ($\mathbf{S}$) sentroidlerin ($\{C_0 C_1 C_{1,döndürülmüş} C_{0,döndürülmüş}\}$) her biri ile karşılaştırılması vasıtasıyla belirlenir. Bu, Şekil 3'te gösterilmiştir;

15 burada bir hedef vektör ($\mathbf{S}$), çıktı olarak bir sınıf göstergesini (C_j) veren bir sınıf atama birimine (302) girilir. $C_{1,döndürülmüş}$ ve $C_{0,döndürülmüş}$ sentroidleri, bir tabloya kaydedilmez, ama sentroidlerin (C_0 ve C_1) elemanlarının döndürülmesiyle "oluşturulur". Elemanların kelimenin tam anlamıyla döndürülmeleri gerekmez; bunun yerine, modifiye edilmiş bir arama işlemi, C_0 , döndürülmüş ve C_1 , döndürülmüş elemanlarını, yani hem sentroidleri hem kod çizelgesi

20 vektörlerini okurken, C_0 ve C_1 elemanlarını ters sırada okuyabilir. Bu şekilde, CB, CB'de fiziksel olarak kayıtlı olana kıyasla iki kat sayıda kod vektörünü içerecek şekilde genişletilebilir; bu, bellek kaynaklarından tasarruf edilmesi anlamına gelir. Yukarıda açıklandığı gibi bu, pik bölgelerinin simetrisinden faydalanılmasından dolayı mümkündür. Daha spesifik olarak, çözüm, bir döndürülmüş geçerli kod vektörünün de bir geçerli kod

25 vektörü olmasından dolayı simetriden faydalanılabileceği gözlemine dayanmaktadır.

Arama bölgesi, girdi hedef vektörlerin sayısına karşılık gelen spektral piklerin sayısına uyarlanmıştır. Bu, piklerin / vektörlerin sayısının bir göstergesini girdi olarak alan ve bir arama bölgesinin bir göstergesini çıktı olarak üreten bir fonksiyonel birimi (402) gösteren Şekil 4'te örneklenmiştir. Şekil 4, ayrıca, arama bölgesi (arama alanı) belirlenirken örneğin

30 VQ'yü uygulayan bir kodekin bit hızının hesaba katılabileceğini betimlemektedir. Örneğin farklı bit hızları için farklı kalite gereksinimleri mevcut olabilir, örneğin bit hızı ne kadar yüksekse beklenen kalite o kadar yüksektir. Ayrıca, örneğin farklı bit hızlarında bir kodekte

eşit ölçüde karmaşık olmayan farklı modüller aktive edildiğinden, izin verilen maksimum karmaşıklık bit hızı ile değişebilir, yani tüm kodlama prosedürüne ilişkin bir maksimum karmaşıklık kısıtlamasından geriye kalan VQ için izin verilmiş karmaşıklık aynı olmayabilir. Yani bit hızı bilgisi, vektör nicelemede dikkate alınabilen kalite ve karmaşıklık koşullarındaki

5 değişiklikleri yansıtır.

Şekil 5'teki tablo, arama bölgesinin piklerin sayısına nasıl uyarlandığını betimlemektedir. Şekil 5'te, arama bölgesi, girdi vektör başına aramadaki katsayıların (kod vektörlerinin) sayısı olarak belirtilmiştir. Şekil 5'teki tablodaki sayılar, Şekil 2'deki CB'nin, iki "normal" veya "fiziksel" ve ikisi "döndürülmüş" veya "sanal" olan dört adet 7 bitlik segmenti (her biri 128

10 kod vektörünü içeren dört segment) içerdiği varsayımı altında türetilmiştir.

Şekil 5'teki tablonun gerisindeki mantık, kodlanacak bir adet daha az pik, örneğin 17 yerine 16 pik mevcut olduğunda "kaydedilmiş" 128 karşılaştırmanın, geriye kalan pikler arasında dağıtılabileceğidir. Bir noktada arama uzunluğu doyuma ulaşır, çünkü CB'nin fiziksel boyutuna ulaşmıştır. Şekil 5'te gösterilen örnekte, piklerin sayısı 8 veya daha az olduğunda bu

15 noktaya ulaşılır. Yani Şekil 5'te gösterilen örnekte, piklerin sayısı 8 veya daha az olduğunda, tüm girdi hedef vektörler (yani pikler) için maksimum izin verilen karmaşıklığa ulaşmadan tam arama yapılabilir.

Arama prosedürünün örnekleri, Şekil 6a-d'de betimlenmiştir. Uygulamada daha önce Şekil 1a-b ile bağlantılı olarak açıklananla aynı tipte aramadır, ama "normal" ve

20 "döndürülmüş" CB segmentlerine / sınıflarına ek sınıflandırmayı içerir.

Şekil 6a'da gösterilen örnekte, girdi vektör (S), C1 sınıfına (aşağı doğru ok ile belirtilen pozisyon) aittir. Bu durumda arama alanı, yalnızca C1 sınıfı ve C_0 ve C_1 sınıflarının ortak alanı arasındaki boyutlar ile sınırlanmıştır. Şekil 6a'da bu, farklı boyutlarda arama alanlarını belirten üç adet kesik çizgili ok ile gösterilmiştir. Şekil 6b, bir girdi vektörün C_0 sınıfına (aşağı

25 doğru ok ile belirtilen pozisyon) ait olduğu durumu göstermektedir; bu durumda arama, farklı bir başlangıç noktasına sahiptir.

Benzer şekilde, Şekil 6c-d'de betimlendiği gibi, eğer girdi vektör bu sınıflardan birine aitse arama, $C_{1,döndürülmüş}$ ve/veya $C_{0,döndürülmüş}$ sınıflarında gerçekleştirilebilir. C_1 ve $C_{1,döndürülmüş}$ 'ün ortak arama alanında arama gerçekleştirilmez. Bunun nedeni, bir normal ve bir döndürülmüş

30 sınıfın ortak alanının, bir gerçek veri kümesine karşılık gelmemesidir (istatistikleri, gerçek veri istatistiklerine karşılık gelmez). Dolayısıyla böyle bir alanda bir girdi vektör için iyi bir eşleşme bulunabilmesi çok olası değildir.

Maksimum izin verilen karmaşıklığın dıştan kontrolünü içeren iletişim sistemi

Hedef vektörlerin (N) sayısına dinamik olarak ayarlanan bir karmaşıklığa sahip bir VQ konsepti, karmaşıklık sınırının önceden belirlenmemiş olduğu duruma genişletilebilir, ama örneğin belirli bir ölçüt bazında değişebilir ve VQ'ye ve/veya VQ'nün uygulandığı antiteye bildirilebilir. Bu, Şekil 7'deki şematik blok diyagramında betimlenmiştir; şekil, piklerin / vektörlerin sayısını veya bunların bir göstergesini girdi olarak alan ve ayrıca bir karmaşıklık kısıtlamasını (L_{MAX}) girdi olarak alan bir fonksiyonel birimi (702) göstermektedir. Fonksiyonel birim (702), CB'nin bir arama alanının / bir bölgesinin bir göstergesini vermektedir; bkz. Şekil 6a-d'deki kesik çizgili oklar.

- 10 Burada sunulan ayarlanabilir karmaşıklığa sahip VQ algoritması, nicelemenin hassasiyeti (yani kalitesi) ve bilgi işlem karmaşıklığının önceden tanımlanmış bir eşik değerin altında tutulması arasında optimal dengeyi sağlar.

CB yapısının elde edilmesi için örnek prosedür

- 15 Bir VQ'de kullanıma yönelik bir CB'nin dizayn edilmesi veya düzenlenmesi için örnek bir prosedür, aşağıda Şekil 8'e referansla açıklanacaktır. Prosedür, bir VQ'de kullanım için örneğin bir MDCT kodlayıcısı gibi bir dönüşüm ses kodlayıcısında niceleme sağlayan bir CB'nin oluşturulmasına yöneliktir.

Aşağıda açıklanan prosedür, bir geleneksel VQ CB oluşturma veya organize etme işleminden ayrılan ve/veya ona ek bir CB oluşturma prosedürünün bölümlerine ilişkindir.

- 20 CB, daha önce açıklandığı gibi, örneğin bir K-ortalamlar algoritmasının kullanımı vasıtasıyla eylem 802'de sınıflara bölünmüştür. Ardından CB'nin kod vektörleri, örneğin denklem (4)'te tanımlanan gibi bir distorsiyon ölçüsü bazında sınıflandırılır. Her kod vektörü için distorsiyon ölçüsü, daha önce açıklandığı gibi kod vektörü ve CB'nin her sınıfını temsil eden sentroidler arasındaki bir ilişkiye bağlıdır.
- 25 CB'nin bu düzenlemesi, arama alanının ve dolayısıyla VQ'deki arama karmaşıklığının, son derece korunan bir VQ kalitesine (örneğin yeniden oluşturulan hedef vektörlerin kalitesine) uyarlanmasını sağlar.

Örnek VQ prosedürü

- 30 Bir vektör niceleyicide (VQ) örnek bir prosedür, aşağıda Şekil 9a'ya referansla açıklanacaktır. Prosedür, örneğin spektral pik bölgelerini kodlayan bir MDCT kodlayıcısı gibi bir dönüşüm ses kodlayıcısında kullanıma uygundur. Ses sinyali, örneğin söz ve/veya müzik içerebilir.

Daha önce açıklandığı gibi bir sayıda (N) girdi hedef vektör, VQ tarafından alınır. Aşağıda, basitlik sağlamak için girdi hedef vektörlerden biri ile ilişkili eylemler açıklanacaktır.

Bir girdi hedef vektör (s), her biri bir CB sınıfını (bkz. daha önce açıklanan sınıflar C_0 ve C_1 , vb.), tercihen her sınıfın sentroidini temsil eden birkaç kod vektörü ile karşılaştırılır.

- 5 Karşılaştırma Şekil 9a-c'de eylem 902 olarak betimlenmiştir. Alternatif olarak eylem 902, Şekil 9c'de betimlenen bir eylem (904) ile entegre edilmiş olarak kabul edilebilir.

Karşılaştırmaların sonucu bazında, girdi hedef vektör (s), eylem 904'te CB'nin sınıflarından birine veya bölümüne atanır. Girdi hedef vektör (s), bir mesafe ölçüsüne (hata ölçüsü) göre kendisiyle en kısa mesafede bulunan, yani kendisine en çok benzeyen sınıfa atanır veya ona
10 ait olduğu sonucu çıkarılır. CB'de aramanın başlangıç noktası, sınıf ataması veya mesafe ölçüsü bazında eylem 906'da belirlenir.

Eylem 910'da kod çizelgesinde bir arama gerçekleştirilebilir. Arama, seçilmiş başlangıç noktasında başlatılır ve bir veya birden fazla sınıfı veya bunların bölümlerini içeren, belirlenmiş bir boyutta olabilen bir arama alanında gerçekleştirilir. Avantajlı şekilde dizayn
15 edilmiş ve düzenlenmiş CB'den dolayı, girdi hedef vektör (s) için tüm CB içindeki tüm aday kod vektörleri için en iyi eşleşmenin arama alanı içinde bulunma olasılığı, arama alanı örneğin CB'nin yarısı ile sınırlı olsa bile çok yüksektir. Arama alanının tüm kod çizelgesini içereceği bir durumda, arama belirlenmiş başlangıç noktasında başlatıldığında en iyi eşleşen kod vektörü, aramada daha erken bulunabilir.

- 20 Belirlenmiş arama alanı içinde en iyi eşleşme bulunduğunda, sonuç olarak, VQ'den, örneğin bir ses kodlayıcısında kullanım için en iyi eşleşen kod vektörünün indeksi eylem 912'de sağlanır.

Ayrıca arama alanının boyutu, Şekil 9c'de gösterilen eylem 908'de belirlenebilir. Arama alanı, girdi hedef vektör (s) için en iyi eşleşmenin aranmasında değerlendirilmesi gereken CB'deki
25 kod vektörlerinin sayısı olarak tarif edilebilir. Arama alanının boyutu, girdi hedef vektörlerin sayısı ve bilgi işlem karmaşıklığına (L_{MAX}) ilişkin bir kısıtlama bazında belirlenir. Dolayısıyla arama alanının boyutu, örneğin nicelenecek sinyalin özelliklerine ve/veya bir kodun niteliklerine bağlı olarak örneğin her kodlama blokunda bir veya bir başka zaman aralığında bir belirlenebilir. Eğer girdi hedef vektörlerin sayısı ve kısıtlama (L_{MAX}) zaman içinde sabitse
30 veya yarı sabitse, arama alanının boyutu da karşılık gelen süre boyunca sabit tutulabilir.

Bu durumda Örnek VQ düzeni

Aşağıda bir dönüşüm kodlayıcısı / kodekinde kullanım için uygun örnek bir VQ düzenlemesi,

Şekil 10'a referansla açıklanacaktır. Dönüşüm kodeki, örneğin bir MDCT kodeki olabilir. VQ, yukarıda açıklanan prosedürün eylemlerini yürütmek üzere uyarlanmıştır.

VQ (1001), bir iletişim birimi (1002) vasıtasıyla başka antiteler (örneğin ses kodeki) ile iletişim kuruyor olarak gösterilmiştir. VQ, ayrıca, örneğin normal fonksiyonları sağlayan fonksiyonel birimler gibi başka fonksiyonel birimleri (1016) içerebilir ve ayrıca bir veya birden fazla depolama birimini (1014) içerebilir.

VQ (1001), örneğin şunlardan biri veya birden fazlası vasıtasıyla uygulanabilir; bir işlemci veya bir mikro işlemci ve uygun depolama birimiyle birlikte uygun yazılım, bir Programlanabilir Mantık Cihazı (PLD) veya başka elektronik bileşen (bileşenler) ve/veya devreler.

İletişim biriminin (1002), örneğin bir kodlama antitesinden sağlanan, girdi hedef vektörler ve L_{MAX} gibi uygun parametrelerin elde edilmesi için fonksiyonel birimleri içerdiği varsayılmıştır.

VQ, bir girdi hedef vektörü (s) CB'nin her sınıfını temsil eden vektörlerle, örneğin her sınıfın sentroid vektörü ile karşılaştırmak üzere uyarlanmış bir karşılaştırma birimini (1004) içerebilir. Ayrıca, VQ, karşılaştırma bazında bir sınıfı girdi hedef vektöre (s) atamak (veya vektörü (s) bir sınıfa atamak, yani vektörün hangi sınıfa ait olduğuna karar vermek üzere uyarlanmış bir atama birimini (1006) içerebilir. Ayrıca VQ, CB'de bir arama için uygun bir başlangıç noktasını vektöre (s) atanmış sınıf bazında belirlemek üzere uyarlanmış bir belirleme birimini (1008) içerebilir. Belirleme birimi, ayrıca, CB'deki bir arama alanını boyutunu, örneğin alınmış girdi hedef vektörlerin bir sayısı ve bir bilgi işlem karmaşıklığı kısıtlaması bazında belirlemek üzere uyarlanabilir.

Ayrıca VQ, CB'de belirlenmiş başlangıç noktasında başlayan ve belirlenmiş arama alanını arayan bir arama gerçekleştirmek üzere uyarlanmış bir arama birimini (1010) içerebilir.

Arama, girdi hedef vektör (s) ile en iyi eşleşen kod vektörünü işaret eden bir veya birden fazla CB indeksi ile sonuçlanmalıdır. VQ, ayrıca söz konusu indeksi veya indeksleri başka bir antiteye, örneğin bir dönüşüm kodekine (veya onun tarafından kullanılması için) sağlamak üzere uyarlanmış bir sağlama birimini (1012) içerebilir.

Örnek düzenek

Şekil 12, örneğin bir dönüşüm ses kodekinde kullanıma uygun, Şekil 5'te gösterilen bir VQ düzenlemesini açıklanan alternatif bir yolu da olabilen bir düzeneğin (1200) bir düzenlemesini şematik olarak göstermektedir. Burada düzenek (1200), örneğin bir DSP'yi (Dijital Sinyal

İşlemcisi) içeren bir eşlim birimini (1206) içerir. İşlem birimi (1206), burada açıklanan prosedürlerin farklı aşamalarını yürütmek için tek bir birim veya çok sayıda birim olabilir. Düzenek (1200), girdi hedef vektörler ve örneğin bit hızının ve/veya karmaşıklık kısıtlamasının göstergeleri gibi sinyallerin alınması için giriş birimini (1202) ve ayrıca en iyi kod vektörleri için CB indeksleri gibi çıktı sinyali (sinyalleri) için çıkış birimini (1204) de içerebilir. Giriş birimi (1202) ve çıkış birimi (1204), düzeneğin donanımında tek bir birim olarak düzenlenebilir.

Bundan başka, düzenek (1200), bir kalıcı bellek, örneğin bir EEPROM, bir flaş bellek ve bir disk sürücü formunda en az bir bilgisayar program ürününü (1208) içerir. Bilgisayar program ürünü (1208), düzeneğin (1200) işlem biriminde (1206) yürütüldüğünde, düzeneğin, daha önce Şekil 9a-c ile bağlantılı olarak açıklanmış bir prosedürün aşamalarını yürütmesine neden olan kod araçlarını içeren bir bilgisayar programını (1210) içerir.

Dolayısıyla açıklanan örnek düzenlemelerde, düzeneğin (1200) bilgisayar programındaki (1210) kod araçları, bir girdi hedef vektörün, bir CB'nin sınıf sentroidleri ile karşılaştırılması için bir karşılaştırma modülünü (1210) içerebilir. Bilgisayar programı, girdi hedef vektöre bir sınıf atamak için bir atama modülünü (1210b) içerebilir. Bilgisayar programı (1210), ayrıca, CB'de bir arama için bir başlangıç noktası belirlemek ve ayrıca girdi parametreler bazında bir arama alanı veya bölgesi belirlemek için bir belirleme birimini (1210c) içerebilir. Bilgisayar programı (1210), ayrıca, CB'nin yukarıdakilere göre aranması için bir arama birimini (1210d) içerebilir. Ayrıca bilgisayar programı (1210), arama biriminden başka antitelere çıkışı yapılan indeksleri sağlamak için bir sağlama modülünü (1210e) içerebilir.

Bilgisayar programı (1210), bilgisayar program modülleri halinde yapılandırılmış bilgisayar program kodu formundadır. Modüller (1210a-e), esasen Şekil 10'da betimlenen VQ'nün (1001) en azından bir kısmını taklit etmek üzere Şekil 9a-c'nin herhangi birinde gösterilen akışın eylemlerini yürütebilir. Başka bir deyişle, farklı modüller (1210a-d) işlem biriminde (1206) yürütüldüğünde, en azından Şekil 10'deki birimlere (1004-1012) karşılık gelirler.

Yukarıda Şekil 12 ile bağlantılı olarak açıklanan düzenlemedeki kod araçları, işlem biriminde yürütüldüğünde düzeneğin ve/veya dönüşüm ses kodlayıcısının, yukarıda belirtilen şekiller ile bağlantılı olarak açıklanan aşamaları yürütmesine neden olan bilgisayar program modülleri olarak uygulanmalarına rağmen, alternatif düzenlemelerde kod araçlarından en az biri, en azından kısmen donanım devreleri olarak uygulanabilir.

Önerilen teknoloji, spesifik örnek düzenlemelere referansla açıklanmasına rağmen, açıklama, genel olarak, yalnızca konsepti betimlemeye yöneliktir ve burada açıklanan teknolojinin kapsamını sınırlıyor olarak değerlendirilmemelidir. Yukarıdaki örnek düzenlemelerin farklı özellikleri, ihtiyaçlara, gerekliliklere veya tercihe göre farklı şekillerde kombine edilebilir.

- 5 Yukarıda açıklanan çözüm, VQ'lerin uygulandığı her yerde, örneğin mobil terminaller, tabletler, bilgisayarlar, akıllı telefonlar, vb. gibi cihazlardaki kodeklerde kullanılabilir.

Etkileşim içindeki birimlerin veya modüllerin seçiminin, bunun yanı sıra birimlerin adlandırılmasının, yalnızca örnekleme amacı taşıdığı ve yukarıda açıklanan usullerden herhangi birinin yürütülmesi için uygun düğümlerin, önerilen işlem eylemlerini yürütebilmek için çok sayıda alternatif şekilde konfigüre edilebileceği anlaşılmalıdır.

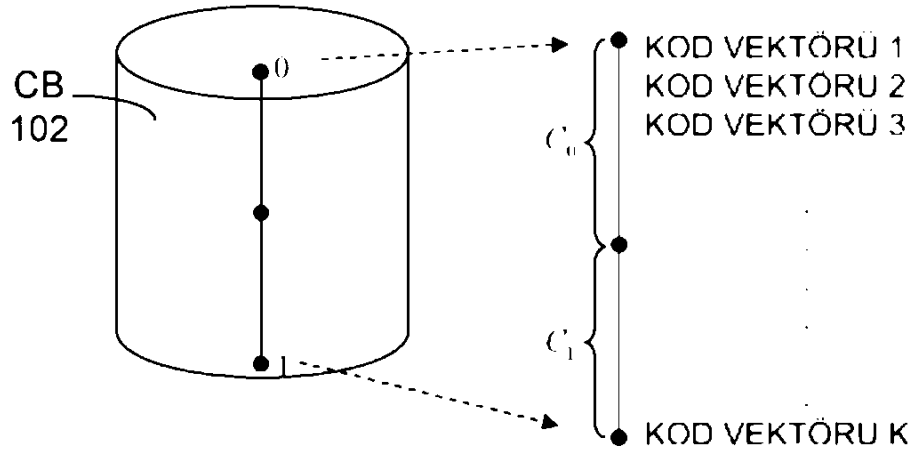
- 10 Sayılanlarla sınırlı olmamak kaydıyla, örneğin "fonksiyonel birim", "işlemci" veya "kumanda birimi" olarak etiketlenenler veya tarif edilenler dahil, fonksiyonel blokları içeren çeşitli elemanların fonksiyonları, devre donanımı ve/veya bilgisayar tarafından okunabilir ortamda kayıtlı kodlanmış komutlar biçimindeki yazılımı yürütebilen donanım gibi donanımın kullanımı vasıtasıyla sağlanabilir. Böylece bu fonksiyonlar ve gösterilen fonksiyonel bloklar, 15 ya donanım uygulamalı ve/veya bilgisayar uygulamalı olarak ve dolayısıyla makine-uygulamalı olarak anlaşılmalıdır.

- Donanım uygulaması açısından fonksiyonel bloklar, sınırlama olmaksızın, dijital sinyal işlemci (DSP) donanımını, azaltılmış komut kümesi işlemcisini, sayılanlarla sınırlı olmamak 20 kaydıyla uygulamaya özgü entegre devreler (ASIC'ler) gibi (örneğin dijital veya analog) donanım devrelerini ve (uygun olduğunda) bu fonksiyonları yürütebilen sonlu otomatları içerebilir veya kapsayabilir.

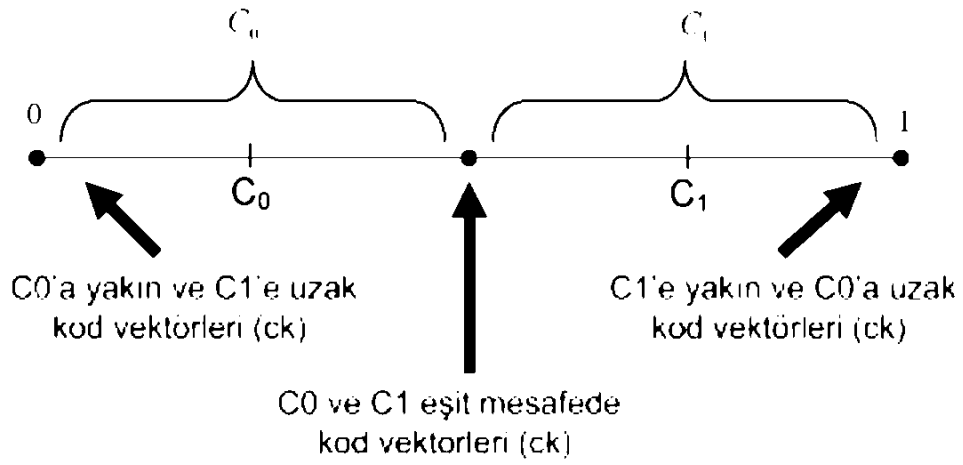
Kısaltmalar

SQ	Skaler Niceleme
VQ	Vektör Niceleme
CB	Kod Çizelgesi
WMOPS	Ağırlıklandırılmış Saniyede Milyon İşlem
MDCT	Modifiye Edilmiş Ayrık Kosinüs Dönüşümü

EP 2 831 757 B1

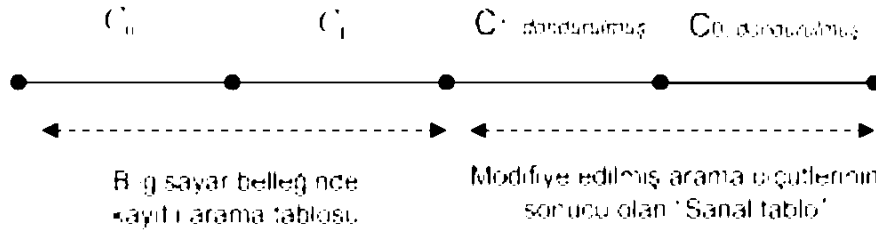


ŞEKİL 1A

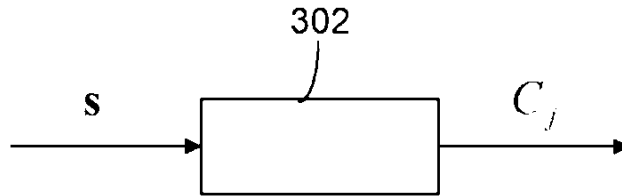


ŞEKİL 1B

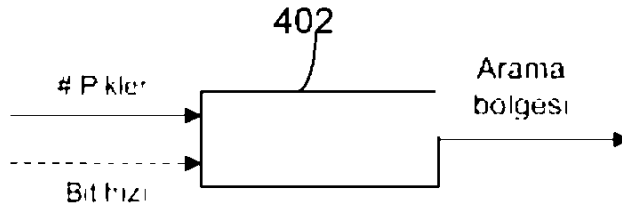
EP 2 831 757 B1



ŞEKİL 2



ŞEKİL 3

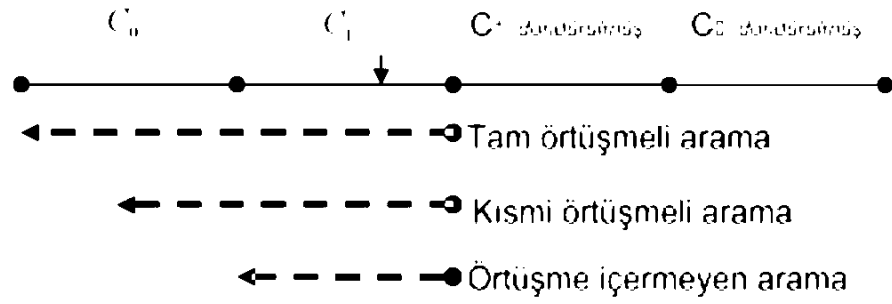


ŞEKİL 4

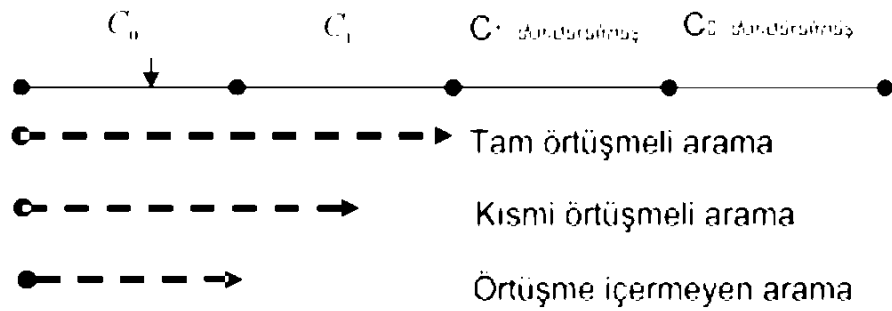
# P kler	17	16	15	14	13	12	11	10	9	8	...	1
# Aramadaki katsayılar	128	136	145	155	167	181	197	217	241	256	...	256

ŞEKİL 5

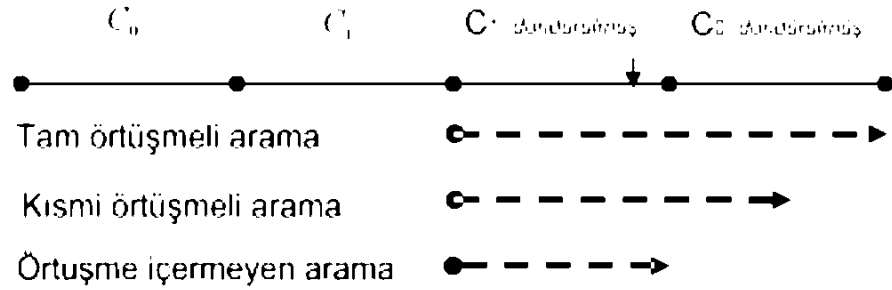
EP 2 831 757 B1



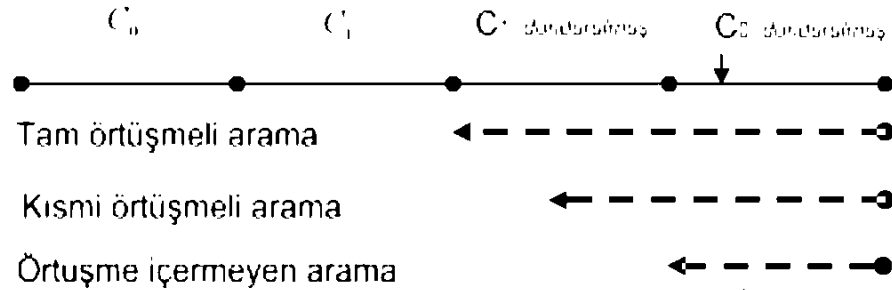
ŞEKİL 6A



ŞEKİL 6B

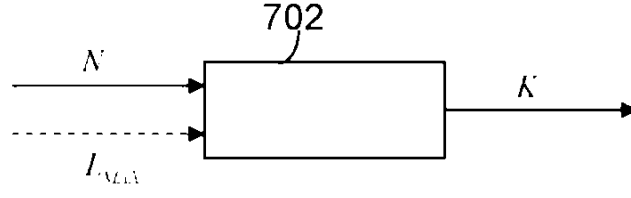


ŞEKİL 6C

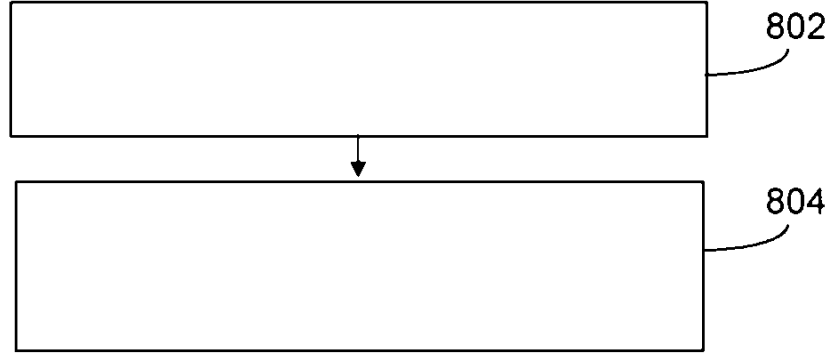


ŞEKİL 6D

EP 2 831 757 B1

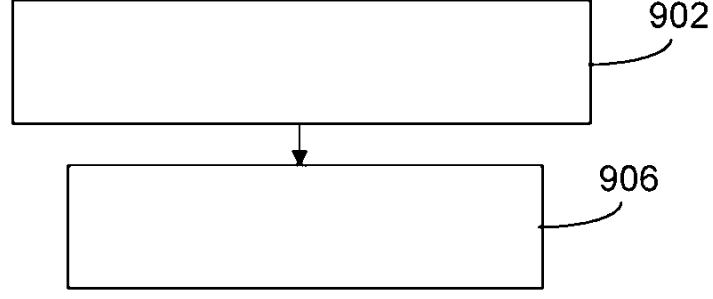


ŞEKİL 7

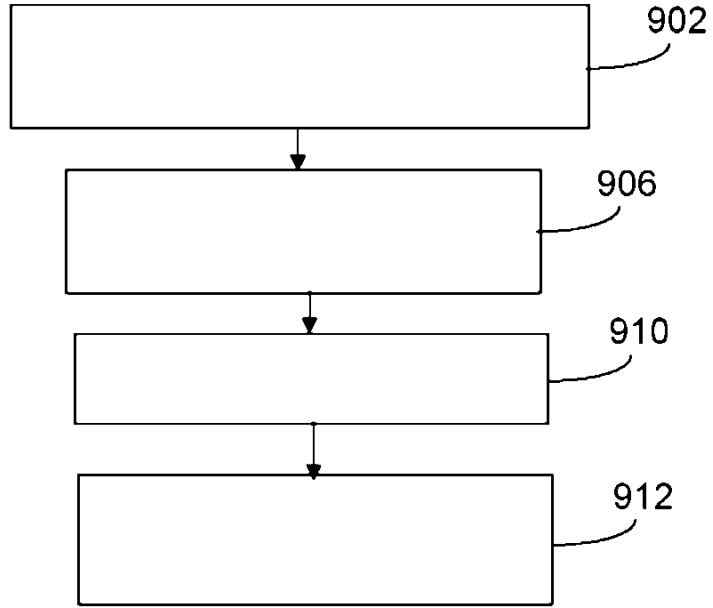


ŞEKİL 8

EP 2 831 757 B1

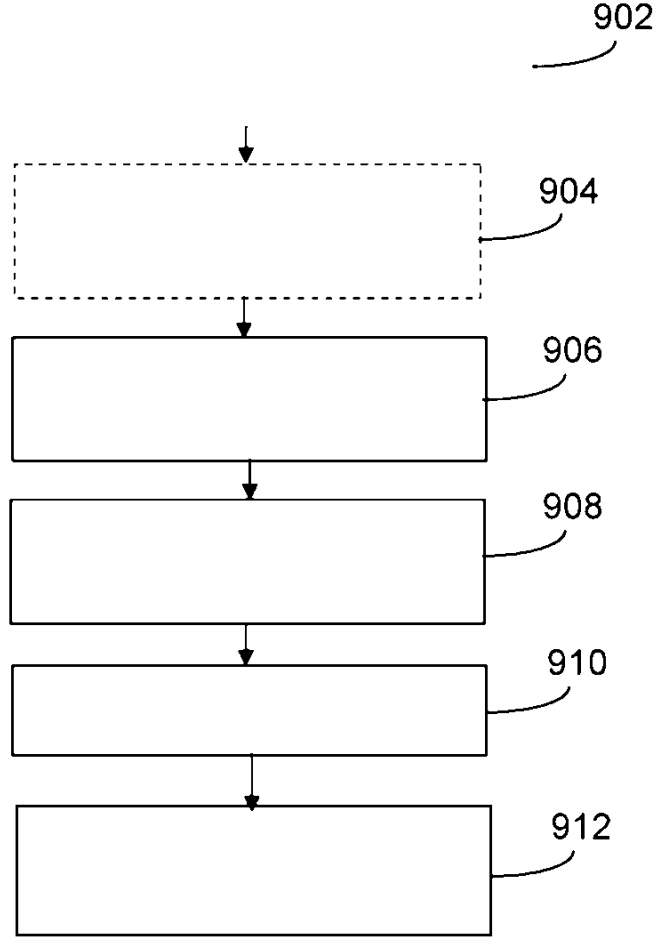


ŞEKİL 9A



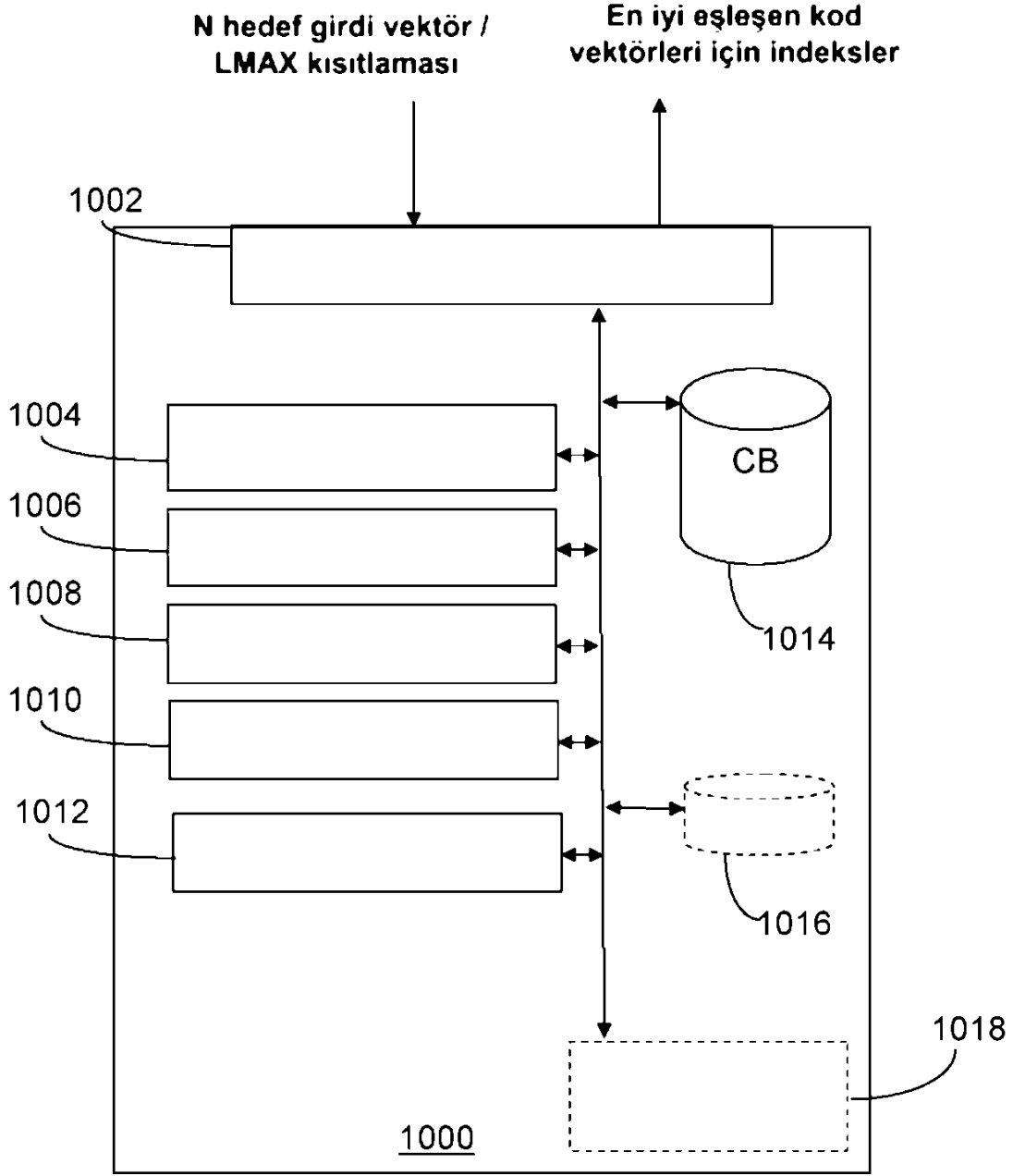
ŞEKİL 9B

EP 2 831 757 B1



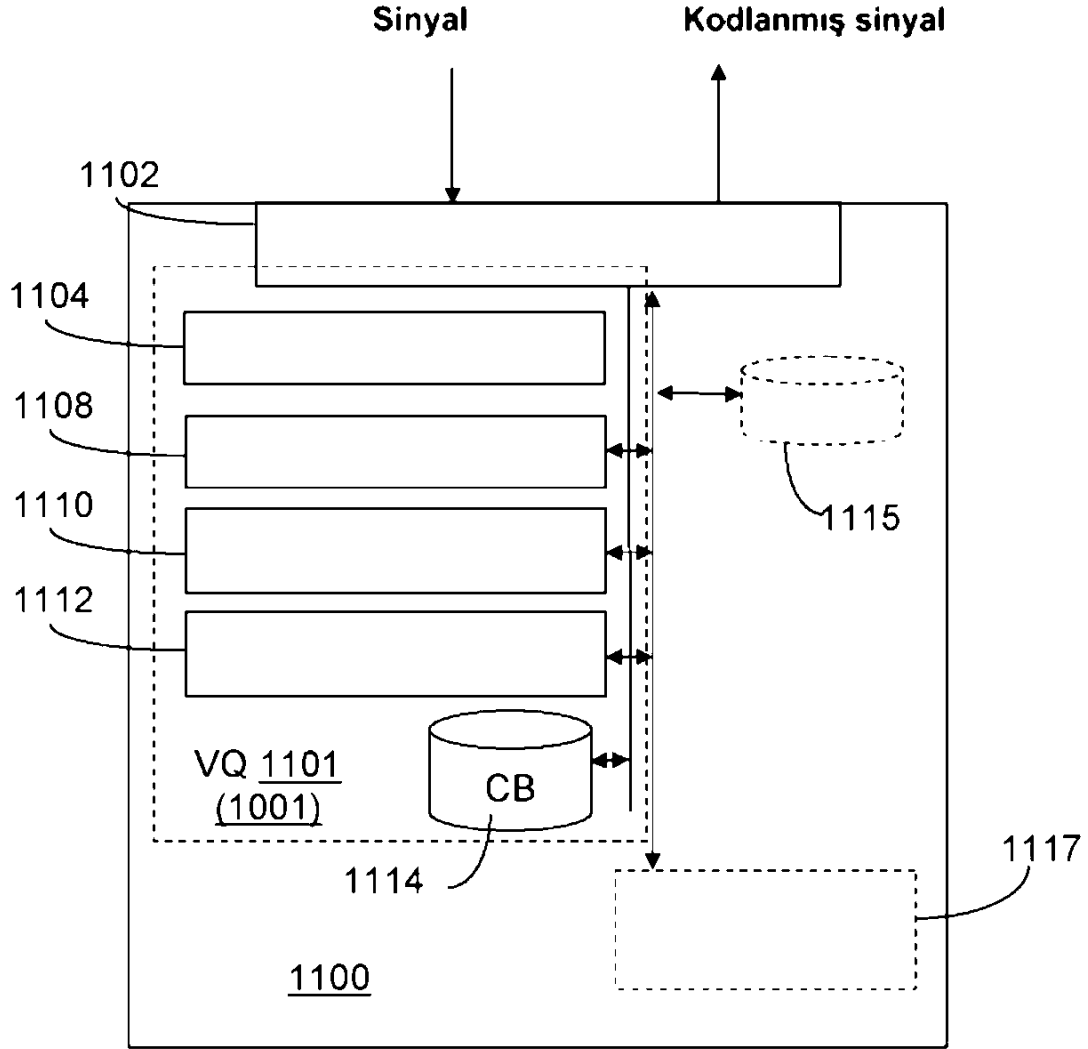
ŞEKİL 9C

EP 2 831 757 B1



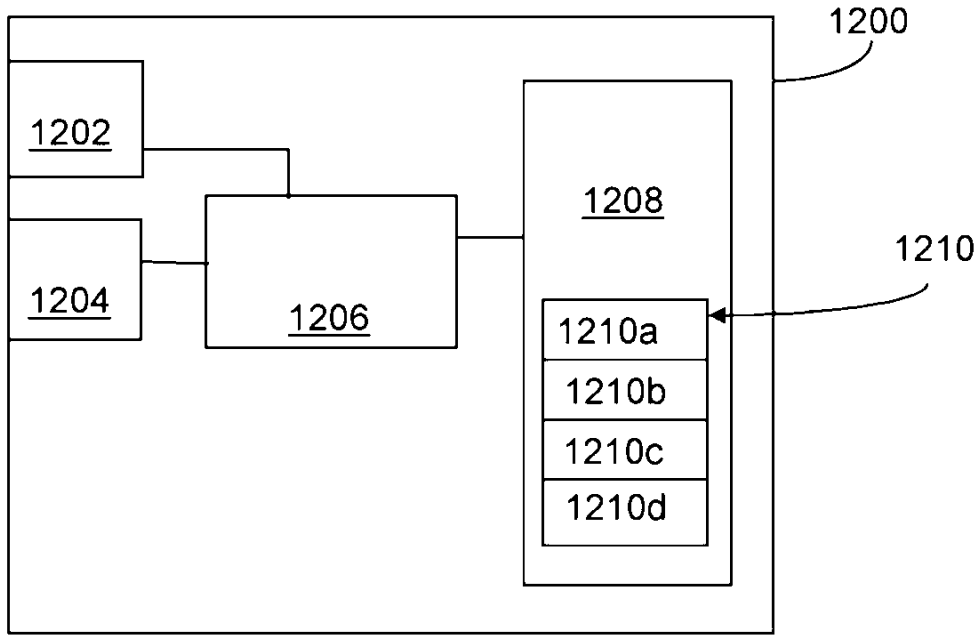
ŞEKİL 10

EP 2 831 757 B1



ŞEKİL 11

EP 2 831 757 B1



ŞEKİL 12