Dalgacık Dönüşümüyle Sıkıştırılmış Videoda Hareketli Bölge Tespiti

Moving Region Detection in Wavelet Compressed Video

B. Uğur Töreyin[1], A. Enis Çetin[1], Anıl Aksay[1], M. Bilgay Akhan[2]

[1]Elektrik ve Elektronik Mühendisliği Bölümü
Bilkent Üniversitesi, Ankara
{ugur,cetin,anil}@ee.bilkent.edu.tr

[2]visiOprime Ltd, 30 St Johns Road, St Johns
Woking, Surrey GU21 7SA, UK
bilgay.akhan@visioprime.com

Özetçe

Görüntü tabanlı pek çok güvenlik sisteminde sıkıştırma yöntemi olarak dalgacık dönüşümü kullanılmaktadır. Bu çalışmada, dalgacık dönüşümüyle sıkıştırılmış videolarda, hareketli nesne tespitiyle ilgili bir yordam geliştirilmiştir. Geliştirilen yordam, her video resmi için arkaplana ait dalgacık dönüşümü katsayılarını önceki video resimleriyle kestirir. Mevcut andaki resme ait dalgacık dönüşümü katsayıları, arkaplana ait katsayılarla karşılaştırılarak, hareketli nesneler tespit edilir. Yordam, özgün resme ait resim öğelerini elde etmek veya arkaplanı kestirmek için dalgacık dönüşümünün tersini almaz. Bu da, halihazırda bulunan hareket kestirimi yöntemleriyle karşılaştırıldığında, hesaplama yönünden daha verimli bir yönteme ve sisteme sahip olunmasını sağlar.

Abstract

In many vision based surveillance systems the video is stored in wavelet compressed form. In this study, an algorithm for moving object and region detection in video that is compressed using a wavelet transform (WT) is developed. The algorithm estimates the WT of the background scene from the WTs of the past image frames of the video. The WT of the current image is compared with the WT of the background and the moving objects are determined from the difference. The algorithm does not perform inverse WT to obtain the actual pixels of the current image nor the estimated background. This leads to a computationally efficient method and a system compared to the existing motion estimation methods.

1. Giriş

Görüntü tabanlı pek çok güvenlik sisteminde, video, yasal sebeplerden dolayı, yalnızca resim içi sıkıştırma teknikleriyle sıkıştırılmaktadır. Mahkemeler, videolardaki öngörülmüş resimleri, yasal bir kanıt olarak kabul etmemektedir [1]. Bu yüzden, güvenlik sistemlerinde kaydedilen videolar genellikle yalnız kendi içinde sıkıştırılmış resimlerden oluşmaktadır. Ayrıca, günümüzde kullanılan çoğu sistem, çeşitli kameralardan gelen görüntüleri doğrudan sıkıştırıp, sabit disklerde saklayacak donanıma sahiptir. Sıkıştırma zorunluluğunun ardındaki temel neden ise kişisel bilgisayarlardaki veri yollarının ham resim bilgisini olduğu haliyle taşıyabilecek sığaya sahip olmamasıdır.

Bu bildiride, kameralardan gelen video verisinin bir biçimde dalgacık dönüşümüyle sıkıştırılmış olduğu kabul edilmiştir. Sıkıştırılmamış video verisi, işlemci ve veri yolu kısıtlamalarından dolayı, gerçek zamanlı olarak çalışan hiçbir sistemde kullanılamamaktadır. Önerilen yöntem, videodaki hareketli nesnelerin tespiti için dalgacık dönüşümünün tersini almayı gereksiz kılmaktadır. Videodaki hareketli bölgeler, mevcut andaki resmin dalgacık dönüşümü katsayılarıyla, önceki resimlere ait dalgacık dönüşümü katsayılarından kestirilen arkaplan dalgacık dönüşümü katsayılarının karşılaştırılması sonucu elde edilmektedir.

Hareketli bölge ve nesneler, mevcut resimle, önceki resimlerle kestirilmiş arkaplan resmine ait dalgacık dönüşümü katsayılarının karşılaştırılmasıyla bulunmaktadır. Eğer iki dalgacık dönüşümü arasında *önemli* bir fark mevcutsa, bu, videoda bir hareket olduğu anlamına gelmektedir. Eğer bir hareket yoksa, mevcut resme ait dalgacık dönüşümüyle arkaplana ait dönüşüm arasında ideal olarak bir fark bulunmamalıdır. Ancak, sıkıştırma işlemi sırasındaki nicemlemeye bağlı olarak iki dönüşüm arasındaki fark, gerçekte sıfır olmamakla birlikte, ihmal edilebilir düzeyde bulunmaktadır.

Arkaplan resmine ait dalgacık dönüşümü katsayıları, diğer katsayılara göre daha durağandır. Bu, arkaplanın zamanla çok da fazla değişmemesinden kaynaklanmaktadır [1-5]. Kamera, görüş açısı içindeki alanı bir süre kaydettiğinde arkaplana ait dalgacık dönüşümü katsayıları kestirilebilir. Çünkü normal olarak videoda hareketli bölgeler, resmin yalnızca bazı bölümlerini kısa süreliğine kapsamaktadır. Bununla birlikte, önplan nesnelerine ait görüntü öğelerinin parlaklık değerleri ve dalgacık katsayıları zamanla değişmektedir. Zamanla önemli ölçüde değişen, yani durağan olmayan dalgacık katsayıları, önplana aittir ve bu katsayılar hareket bilgisi içermektedir.

Arkaplana ait dalgacık dönüşümünün kestirilmesi için önerilebilecek en basit yöntem, mevcut ana dek kaydedilmiş tüm resimlere ait dönüşümlerin ortalama değerlerinin alınması olabilir. Hareketli bölgeler, resmin yalnızca bir bölümünü, bir süreliğine kapatacağı için, ortalama resimdeki etkileri zamanla yok olacaktır.

0-7803-8318-4104/$20.00©2004 IEEE

Gerçek zaman başarımı sağlandığı müddetçe, arkaplan kestirimi için herhangi bir uzam alanı yöntemi [2-7] dalgacık alanında da gerçekleştirilebilir. Örneğin, [2]'de önerilen yöntem, arkaplan kestirim denkleminin her iki tarafının dalgacık dönüşümü alınarak gerçekleştirilebilir.

2. Hareketli Nesne Tespiti İçin Karma Bir Yöntem

Arkaplan çıkarımı, resimde ilgilenilen nesnelerin ayrıştırılması için, güvenlik uygulamalarında da yaygın olarak kullanılan bir yöntemdir. Konuyla ilgili olarak yazında değişik birçok çalışma bulunmaktadır [1-5]. Örneğin, [2]'de bahsedilen arkaplan kestirimi yöntemi, her görüntü öğesine, bağımsız olarak basit bir sonsuz dürtü yanıtlı süzgeç uygulanmasından ibarettir. Böylelikle arkaplan resmi güncellenmektedir. Bu arkaplan resmiyle, benzer şekilde güncellenen eşik değerleri, ayrı ayrı herbir görüntü öğesini önplan görüntü öğesi veya arkaplan görüntü öğesi olarak sınıflandırmak için kullanılmaktadır.

Videodaki durağan, parlaklık değerleri değişmeyen görüntü öğeleri, arkaplana ait olanlardır, çünkü arkaplan, videonun zaman açısından durağan olan kısmı olarak tanımlanabilir. Eğer kameranın görüntülediği alan bir süre kaydedilirse, tüm arkaplana ait görüntü öğeleri kestirilebilir. Bunun nedeni, kaydedilen videodaki hareketli bölgelerin ve nesnelerin, normal olarak resimlerin belli bir kısmını kapsamasındandır. Arkaplan kesitirimi için en basit yöntem, eldeki tüm resimlerin ortalamasının alınmasıdır. Hareketli nesneler resmin yalnızca bir bölümünü kapsadığı için, bu bölgelerin arkaplan resmindeki etkileri ortalama sayesinde zamanla kaybolacaktır.

Arkaplan kestirimi için, [2]'de, özyinelemeli bir yordam öne sürülmüştür. $I_n(x,y)$'nin *n.* görüntüde, *(x,y)* konumundaki görüntü öğesine ait parlaklık değerini temsil ettiğini varsayalım. Şu halde, aynı konuma ait olarak kestirilen arkaplan parlaklık değeri, $B_{n+1}(x,y)$, aşağıdaki gibi hesaplanır:

$$B_{n+1}(x,y)=\begin{cases} aB_n(x,y)+(1-a)I_n(x,y), & (x,y)\ hareketli\ değil \\ B_n(x,y), & (x,y)\ hareketli \end{cases} \quad (1)$$

Yukarıda geçen $B_n(x,y)$ ise, aynı konumdaki arkaplan görüntü öğesinin bir önceki parlaklık kesitirim değeridir. Aynı denklemdeki *a* güncelleme parametresi ise bire yakın gerçek bir sayıdır. Yordamın başında, $B_0(x,y)$ arkaplan değeri, $I_0(x,y)$ ilk resim değerini alır.

Herhangi bir *(x,y)* konumundaki görüntü öğesi, eğer bu öğeye ait I_n ve I_{n-1} resimlerindeki parlaklık değerleri aşağıdaki eşitsizliği sağlıyorsa, hareketli olarak kabul edilmektedir:

$$|I_n(x,y) - I_{n-1}(x,y)| > T_n(x,y) \quad (2)$$

Burda geçen $I_{n-1}(x,y)$, *(x,y)* konumundaki resim öğesinin *(n-1).* resimdeki parlaklık değeridir. $T_n(x,y)$ ise, *(x,y)* konumundaki görüntü öğesine ait, istatistiksel olarak *önemli* bir parlaklık değişimine karşılık gelen, eşik değeridir. Bu eşik değeri de, her görüntü öğesi için ayrı ayrı, özyinelemeli olarak şu şekilde güncellenmektedir:

$$T_{n+1}(x,y)=\begin{cases} aT_n(x,y)+(1-a)(c|I_n(x,y)-B_n(x,y)|), & (x,y)\ hareketli\ değil \\ T_n(x,y), & (x,y)\ hareketli \end{cases} \quad (3)$$

Bu denklemdeki *c* birden büyük, *a* güncelleme parametresi ise bire yakın bir gerçek sayıdır. Eşik değişkenlerinin ilk değerleri deneysel olarak belirlenen bir sayıya eşitlenmiştir.

Son denklemde görüldüğü gibi, *c* parametresi ne denli büyükse, eşik değerleri de o ölçüde büyük olmaktadır. Bu da hareket tespiti hassasiyetini o denli düşürmektedir.

Arkaplandan *önemli ölçüde* farklı olan bölgeler hareketli olarak kabul edilmiştir. Bu kabule göre, hareketli bölgeler, kestirilen arkaplanla mevcut andaki resim arasındaki farktan elde edilmiştir. Resim öğesi bazında aşağıdaki eşitsizliği sağlayan tüm öğeler hareketli olarak tespit edilmiştir:

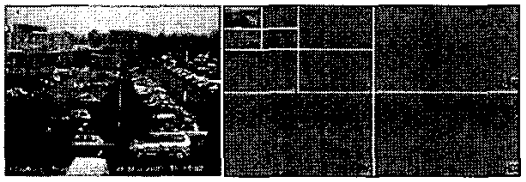
$$|I_n(x,y) - B_n(x,y)| > T_n(x,y) \quad (4)$$

Herhangi bir *(x,y)* konumundaki resim öğesi (4)'ü sağladığı sürece hareketli nesne öğesi olarak belirlenmektedir.

3. Dalgacık Alanında Hareket Tespiti

Yukarda öne sürülen yöntem ile, [6] ve [7]'de öne sürülenler, sıkıştırılmış alanda da aynen uzam alanındaki gibi geçerlidir [3]. [3]'te, videoda hareket tespiti için ayrık kosinüs alanı verisi kullanılmaktadır. Öte yandan bizim çalışmamızda, veri sıkıştırılması için dalgacık dönüşümüne dayalı bir kodlayıcı kullanılmıştır.

Arkaplana ait dalgacık dönüşümü, geçmiş resimlere ait dalgacık dönüşümleri kullanılarak kestirilebilir. Arkapalana ait dalgacık dönüşümü katsyıları, zamanla çok değişmediklerinden durağan bir karakterdedirler. Zaman içinde durağan olmayan katsayılar ise, önplana ait katsayılar olup hareket bilgisine sahiptirler. Eğer kamera bir süre kayıt yaparsa, tüm arkaplana ait dalgacık katsayıları kestirilebilir, çünkü görüntüdeki hareketli bölgeler, resmin ancak bir kısmını, bir süreliğine işgal etmektedir.



Şekil 1: Özgün resim ve bu resme ait altbant resimlerini içeren üç seviye dalgacık ağacı (parlaklık bilgisi gösterilmiştir).

Herhangi bir *n* anındaki B_n arkaplanına ait bir altbant resmi D_n olsun. D_n'nin, Şekil 1'de gösterilen altbant resimlerinden herhangi birini temsil edebildiğini varsayalım. Bu altbant resmiyle kestirilen resme D_{n+1} dersek, D_{n+1} şu şekilde hesaplanır:

$$D_{n+1}(i,j)=\begin{cases} aD_n(i,j)+(1-a)J_n(i,j) & (i,j)\ hareketli\ değil \\ D_n(i,j) & (i,j)\ hareketli \end{cases} \quad (5)$$

Burda J_n, mevcut andaki I_n resminin ilgili altbant resmine karşılık gelmektedir. Güncelleme parametresi *a* bire yakın gerçek bir sayıdır. Arkaplana ait ilk altbant resmi, D_0, videonun ilk resminin, I_0, ilgili altbant resmine eşitlenmiştir.

(1)'den (4)'e dek tüm ifadelerde geçen *(x,y)*'ler, özgün resimdeki görüntü öğeleri konumlarına karşılık gelmektedir. Halbuki, (5) ve bu bölümdeki tüm ifadelerde geçen *(i,j)*'ler

ise, altbant resimlerindeki dalgacık dönüşümü katsayılarının konumlarına karşılık gelmektedir.

Herhangi bir altbant resminde *(i,j)* konumundaki dalgacık dönüşümü katsyısı, aşağıdaki şartı sağlıyorsa hareketli kabul edilmektedir:

$$|J_n(i,j) - J_{n-1}(i,j)| > T_n(i,j) \quad (6)$$

Burdaki $T_n(i,j)$, aşağıdaki gibi özyinelemeli olarak her dalgacık katsayısı için güncellenmektedir:

$$T_{n+1}(i,j) = \begin{cases} aT_n(i,j) + (1-a)(c|J_n(i,j) - D_n(i,j)|) & , (i,j) \text{ hareketli değil} \\ T_n(i,j) & , (i,j) \text{ hareketli} \end{cases} \quad (7)$$

Yukarda geçen *b* birden büyük, *a* ise bire yakın birer gerçek sayıdır. İlk eşik değerleri deneysel olarak bulunabilir.

Arkaplandan yeterince farklı olan bölgeler hareketli olarak kabul edildiği için, bu şekilde kestirilen arkaplana ait altbant resimleriyle, mevcut andaki resme ait altbant resimleri arasındaki fark sayesinde hareketli dalgacık katsayıları ve böylelikle hareketli nesneler tespit edilir. Bir başka deyişle, aşağıdaki eşitsizliği sağlayan dalgacık katsayıları hareketli katsayılar olarak tespit edilir:

$$|J_n(i,j) - D_n(i,j)| > T_n(i,j) \quad (8)$$

Bu eşitsizliği sağlayan tüm dalgacık katsyıları tespit edildikten sonra, bu katsayıların özgün resimde karşılık geldiği konumlar belirlenir. Eğer, sıkıştırma için tek seviye Haar dalgacık dönüşümü kullanıldıysa, (8) eşitsizliğini sağlayan dalgacık katsayısı, özgün I_n resminde ikiye ikilik bir bloğa karşılık gelir. Örneğin, I_n resmi için elde edilen $HH_n(1)$ altbant resminde (veya diğer $HL_n(1)$, $LH_n(1)$, $LL_n(1)$ altbant resimlerinde), *(i,j)* konumundaki dalgacık katsayısı (8)'i sağlıyorsa, özgün resimde bu durum, ikiye ikilik bir resim öğesi bloğundaki harekete karşılık gelir ki bu blok özgün resimdeki şu görüntü öğelerini kapsar: $I_n(k,m)$, $k=2i,2i-1$ ve $m=2j,2j-1$. Tek bir katsayının, dört elemanlı bir bloğa karşılık gelmesinin sebebi, ayrık dalgacık dönüşümü hesaplanması sırasında kullanılan seyrek örnekleme işlemidir. Benzer şekilde, $HH_n(2)$ altbant resminde (veya diğer $HL_n(2)$, $LH_n(2)$, $LL_n(2)$ ikinci seviye altbant resimlerinde), *(i,j)* konumundaki dalgacık katsayısı (8)'i sağlıyorsa, özgün resimde bu durum, dörde dörtlük bir görüntü öğesi bloğunun hareketine karşılık gelir, $I_n(k,m)$, $k=4i,4i-1,4i-2,4i-3$ ve $m=4j,4j-1,4j-2,4j-3$. Genellemek gerekirse, herhangi bir *l.* seviye dalgacık katsayısındaki değişim, 2^l'ye 2^l'lik bir özgün resim bölgesindeki harekete karşılık gelmektedir.

Visioprime şirketi, sıkıştırılmış video verisini sistemimize Aware şirketinin geliştirdiği "Motion Wavelet" biçiminde besleyen, yazılım ve donanımdan oluşan bir video işleme sistemi tasarlamıştır [8]. Sistemimize gelen veri, Daubechies'nin 9/7 çiftdikgen dalgacık dönüşümü kullanılarak elde edilen katsyıları içeren bir biçimdedir. Bu çiftdikgen dönüşümde, herbir dalgacık katsayısının hesaplanmasına dörtten fazla görüntü öğesi katılmaktadır. Ancak katkının önemli bir bölümü görüntü öğesinin hemen yakınındaki komşuluğundan gelmektedir. Birinci seviye ayrıştırma için bu komşuluk, $I_n(k,m)=(2i,2j)$ öğelerini, *l.* seviye içinse $I_n(k,m)=(2^li,2^lj)$ öğelerini içermektedir. Bu sebeple, çalışmamızda hareketli özgün resim öğelerinin belirlenmesi esnasında, hemen yakındaki komşuluk olarak, örneğin *l.* seviye dalgacık ayrıştırması için, $(2^li,2^lj)$ konumundaki öğeyi kapsayan bloğu aldık.

Hareketli görüntü öğelerini özgün resimde yukarda anlatılan biçimde tespit ettikten sonra, bu öğelerin birleşimi alınarak hareketli nesneler videoda konumlandırılır. Bu görüntü öğelerine, bir bölge büyütme yordamı sayesinde hemen komşuluklarındaki resim öğeleri de eklenir. Bölge büyütme yordamı, bu öğeler için aşağıdaki şartın sağlanıp sağlanmadığını denetler:

$$|J_n(i+m,j+m) - D_n(i+m,j+m)| > KT_n(i+m,j+m) \quad (9)$$

Burda geçen $m=-1,+1$, ve $0.8<K<1$, $k \in \mathbb{R}$.

K parametresinin değerini değiştirerek, komşu görüntü öğelerinin eşik değerleri düşürülür, böylelikle bu öğeler de hareketli bölgeler içine dahil edilir.

Bu sınıflandırmadan sonra, artık hareketli nesneler oluşturulur ve bu nesneler, kendilerini çevreleyen en küçük kutularıyla işaretlenir.

4. Test Sonuçları

Yukarda açıklanan, bu çalışmamızda geliştirdiğimiz yordamı Microsoft Visual C++ 6.0'da kodlayıp, Windows XP işletim sistemi altında, 1500 MHz Pentium 4 işlemcili bir kişisel bilgisayarda çalıştırdık. Kişisel bilgisayar tabanlı sistemimiz, gereçek zamanlı olarak 16 ayrı kameradan gelen 5 resim/saniye'lik videoyu işleyebilmektedir.

Yöntemimizin başarımı, resim sayısı 150 ile 3625 arasında değişen 65 farklı video dizisinde sınanmıştır. Bu videolar, senaryolarının çeşitliliği, ışık değişimleri, hareketli nesne boyutlarının farklılığı ve hem içerde, hem dışarda çekilen dizilerin bulunması yönleriyle, bize, uygun başarım değerlendirme olanağı sağlamıştır.

Testlerimizde, karşılaştırma amacıyla, hareketli nesneleri öncelikle özgün resme ait görüntü öğelerini kullanarak bulduk. Daha sonra, yalnız altbant resimlerini kullanan, geliştirdiğimiz yöntemle hareketli nesneleri tespit ettik.

Denenen tüm video dizilerinde bulunan hareketli bölgeler, özgün resme ait görüntü öğeleri kullanılarak tespit edildiklerinde, altbant yöntemiyle bulunan bölgelere göre, uzamsal olarak daha kesin sonuçlar verdi. Bu da beklediğimiz bir neticeydi, çünkü dalgacık piramidinde üst seviyelere çıktıkça, yani, kullandığımız veri, dalgacık dönüşümünün yalnızca üst seviye katsyılarını kapsadıkça, uzamsal kesinlikte düşüş olmaktadır. Bununla birlikte, hareketli bölgelerin tespit başarımı, her iki yöntemde de aynı çıkmıştır. Şekil 2 ve 3 iki farklı hareket tespiti örneğini karşılaştırmalı olarak göstermektedir.

Şekil 2: Özgün resme ait görüntü öğeleriyle bulunan bölgeler (solda) ve yalnızca altbant verisi kullanılarak bulunan bölgeler (sağda).

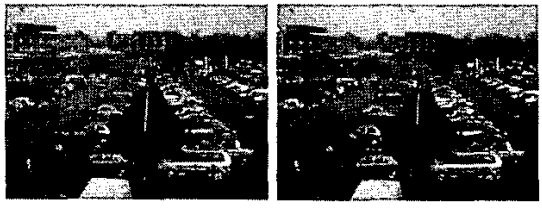
Testlerimiz sonucunda şunu gördük ki, eğer videodaki hareketli nesnelerin boyutları ve aralarındaki

uzaklık, tüm resim çerçeve boyutunun yaklaşık %5'i civarındaysa, iki yöntemle bulunan bölgeler hemen hemen birbirine yakın çıkmaktadır. Şekil 2'de gösterilen, buna örnek bir durumdur.

Hareketli nesnelerin boyutları küçüldükçe ve nesneler birbirine yaklaştıkça, altbant verisi kullanılarak bulunan bölgeler birleşmektedir.

Şekil 3: Arabanın yanındaki adam, özgün resme ait görüntü öğeleri kullanıldığında ayrı bir nesne olarak bulunur (solda). Yalnız altbant verisiyle hareketli nesne tespit yöntemiyle, arabanın yanındaki adam, gruptan ayırtedilememektedir (sağda).

Bu *çözünürlük* sorunu dışında, 65 dizinin tümünde, her iki yöntem de aynı bölgeleri işaretlemektedir. Karanlık ve önplanla arkaplanın yakın parlaklık değerlerine sahip olduğu videolarda, her iki yöntem de benzer sorunlarla karşılaşmaktadır. Örneğin ikisi de kimi hareketli nesneleri tespit edememektedir. Ancak bu sorun özgün resme ait görüntü öğelerine dayalı yöntemde de olduğu için, geliştirdiğimiz sistem için bir götürü teşkil etmemektedir. Bu duruma da bir örnek Şekil 4'te gösterilmektedir.



Şekil 4: "Otopark" dizisinin 181. resminde, en uzakta tespit edilen araba (sağda), dizinin 180. resminde (solda) tespit edilememiştir. Bu örnekteki hareket tespitinde yalnız altbant verisi kullanılmıştır.

5. Sonsöz

Bu çalışmada, dalgacık dönüşümüyle sıkıştırılmış videoda, ters dalgacık dönüşümü almaksızın, yalnız altbant verisi kullanarak bir hareketli nesne tespiti yöntemi geliştirdik. Test sonuçlarımız göstermektedir ki, hareket tespiti açısından başarım, özgün resme ait görüntü öğesi değerleri kullanılarak geliştirilen hareket tespiti yöntemlerinin başarımlarıyla aynıdır.

Eğer nesneler ve aralarındaki uzaklık yeterince büyükse, hareketli bölge bilgisi kullanılarak, bulunan nesnelerin takibi bile mümkün görülmektedir.

Önerdiğimiz yöntemi, yaygın olarak kullanılan diğer yöntemlere üstün kılan tarafı, diğerlerine göre işlemlerinde çok daha az veri kullanmasından dolayı sağladığı hesaplama kolaylığının yanında, video işleme sistemlerindeki kısıtlı bant genişliği sorununu da çözmesidir. Örneğin, 16 kanallı bir video sisteminden aralıksız gelen özgün görüntü elemanı bilgisini, sıradan bir kişisel bilgisayarın PCI veri yoluna gerçek zamanda beslemek mümkün değildir. Bununla birlikte, 16 kanaldan gelen sıkıştırılmış video bilgisini gerçek zamanda PCI veri yoluna besleyip, önerdiğimiz yöntem ve geliştirdiğimiz sistemle sıradan kişisel bilgisayarlarda hareketli nesne tespiti yapmak mümkündür.

6. Kaynakça

[1] G.L. Foresti, P. Mahonen, C.S. Regazzoni, *Multimedia Video-Based Surveillance Systems: Requirements, Issues and Solutions*, Kluwer, 2000.

[2] R.T. Collins, A.J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, L. Wixson "A System for Video Surveillance and Monitoring: VSAM Final Report" *Tech. Report CMU-RI-TR-00- 12*, Carnegie Mellon University, May 2000.

[3] I.B. Ozer, W. Wolf, "A Hierarchical Human Detection System in (Un)Compressed Domains", IEEE Transactions on Multimedia, pp. 283-300, June 2002.

[4] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Who, When, Where, What: A Real Time System for Detecting and Tracking People", Third Face and Gesture Recognition Conference, pp. 222-227, April 1998.

[5] M. Bagci, Y. Yardimci and A.E. Cetin, "Moving Object Detection Using Adaptive Subband Decomposition and Fractional Lower Order Statistics in Video Sequences", Elsevier, *Signal Processing*, pp. 1941-1947, December 2002.

[6] Naoi et al., "Image Processing Apparatus", *U.S. Patent* 6,141,435, October 31st, 2000.

[7] Taniguchi, "Moving Object Detection Apparatus and Method", *U.S Patent*. 5,991,428, November 23rd, 1999.

[8] Motion Wavelets, http://www.aware.com/compression