

Gömülü Sistemler için Hızlı ve Hassas Bir Hareket Belirleme Algoritması

A Fast and Accurate Motion Detection Algorithm for Embedded Systems

N. Cihan Camgöz, Z. Cihan Tayşi, M. Amaç Güvensan, M. Elif Karşılıgil, A. Gökhan Yavuz
Akıllı Sistemler Laboratuvarı, Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, İstanbul, Türkiye
Email: ncihan,cihan,amac,elif,gokhan@ce.yildiz.edu.tr

Özetçe —Son yıllarda Telsiz Algılayıcı Ağlar konusunda hızlı gelişmeler yaşanmıştır. Özellikle yakın zamanda, CMOS kamera ve mikrofonlar gibi donanımların maliyetlerinin azalmasıyla Telsiz Çokluortam Algılayıcı Ağların (TCAA) gelişmesinde önemli rol oynamıştır. Bu ağların geniş uygulama yelpazesinde, kapalı ortam gözetim sistemleri de yer almaktadır. Hemen her gözetim sisteminin ilk adımı hareket belirleme işlemidir. Sistemde hareketin belirlenmesiyle, hareketli bölgeler daha fazla işlenmek üzere diğer adımlara aktarılmakta veya baz istasyonuna ortamda hareket olduğu uyarısı gönderilmektedir. Bu çalışmada, arka planın sürekli hareketli olmadığı kapalı ortam gözetim sistem senaryolarında kullanılmak üzere bir hareket belirleme algoritması tasarlanmış ve mevcut yöntemlerle karşılaştırılmıştır. Goyette'nin veri setleri kullanılarak yapılan testler sonucunda, önerdiğimiz algoritmanın, 97% doğruluk oranı ve ARM9 platformunda QVGA çözünürlükte 48ms'lik ortalama tamamlanma süresiyle, arka planın stabil olduğu senaryolar için mevcut yöntemlerden daha başarılı olduğu gözlenmiştir.

Anahtar Kelimeler—Telsiz Çokluortam Algılayıcı Ağlar, Kapalı Ortam Gözetim Sistemleri, Hareket Belirleme

Abstract—Last decade witnessed the rapid development of Wireless Sensor Networks. More recently, the availability of inexpensive hardware such as CMOS cameras and microphones that are able to ubiquitously capture multimedia content from the environment has fostered the development of Wireless Multimedia Sensor Networks (WMSNs). There is a wide range of applications that are using Wireless Multimedia Sensor Networks, including Indoor Surveillance Systems. Nearly all surveillance systems start with a motion detection algorithm. After detection of motion in an image, either the motion areas are sent to another algorithm for more processing or an alarm is sent to the base station informing that there is motion in the environment. In this paper, we proposed a new motion detection algorithm, which is specifically designed for scenarios with no constant movement in the background. Our tests using Goyette's datasets show that, our proposed algorithm achieved a 97% accuracy with an average execution time of 48ms for QVGA images on ARM9 architecture, and thus outperformed the two currently available methods.

Keywords—Wireless Multimedia Sensor Networks, Indoor Surveillance Systems, Motion Detection

I. GİRİŞ

Telsiz Çokluortam Algılayıcı Ağlar, gömülü bilgisayar sistemlerindeki ilerlemeler ve görsel algılayıcılar gibi pahalı olmayan çokluortam algılayıcılarının kolay erişilebilirliği sayesinde oldukça yaygınlaşmıştır [1]. TCAA'lar, buldukları ortamdan kısıtlı sayısal veriler toplayabilen geleneksel Telsiz Algılayıcı Ağların aksine, işitsel vb. bilgi elde etme yeteneğine sahiptir. Görsel bilgi elde etmenin sağladığı avantajlar sayesinde, kapalı ve açık ortam gözetimi, trafik denetimi, gelişmiş sağlık hizmetleri ve endüstriyel işlem kontrolü gibi görevleri yerine getiren bir çok TCAA uygulaması günlük hayatta kullanılmaktadır [2].

TCAA'nın en yaygın kullanım alanlarından biri, kapalı ve açık alan gözetim sistemleridir. Telsiz haberleşme, algılayıcıların kablolardan bağımsız olarak konumlanmasına olanak sağlarken haberleşme sırasında işlemciyi meşgul etmekte ve kısıtlı pil ömrünü tüketmektedir. Ayrıca görsel verilerin kullanıldığı bu sistemlerde aktarılan verilerin büyük olması [3], ağ trafiğinde kapasite yetersizliği vb. problemlere yol açmaktadır. Bu nedenlerle TCAA'larda sadece anlamlı bilgilerin baz istasyonuna transferi önem kazanmaktadır. Öte yandan, yeni nesil mikroişlemcilerin işlem kapasitesi ve güç tasarrufu konularında gelişmesi, araştırmacıları, kodlama, sıkıştırma, nesne algılama vb. sinyal işleme algoritmalarını düğümlerde gerçekleştirmek konusunda teşvik etmektedir. Bu gelişmeler sonucunda, yerinde işleme (in-situ processing) TCAA'da ön plana çıkmıştır [4].

Kapalı ve açık alan gözetim sistemlerinde bir çerçevenin işlenmeye değer olmasının başlıca koşulu o çerçevede hareketin algılanmasıdır. Bir çerçevede hareket olup olmadığını anlamının en basit yolu, yeni gelen çerçevenin, içinde hareketin olmadığı, arka plan çerçevesi olarak adlandırılan, daha eski bir referans çerçevesiyle karşılaştırılmasıdır. Çerçevede hareket olduğu algılanırsa, sistemin gereksinimlerine bağlı olarak çerçeve üzerinde işlem yapılmaya devam edilir veya kullanıcı bilgilendirmek üzere çerçeve diğer düğümler üzerinden baz istasyonuna gönderilir. Bu çalışmada, TCAA yardımıyla, kapalı bir ortamda dolaşan yabancı kişilerin belirlenmesi amacıyla tasarlanan, daha az işlem yükü ve daha yüksek başarı oranına sahip yeni bir hareket algılama algoritması tanıtılmaktadır. Hareketin sınırlarının belirlenmesiyle, takip eden yüz algılama ve tanıma algoritmaları

gibi işlem yükü ağır algoritmaların daha az alana uygulanarak toplam işlem yükünün azaltılması ve güçten tasarruf edilmesi hedeflenmektedir.

Bu çalışmada, öncelikle mevcut hareket algılama algoritmaları derinlemesine incelenmekte ve benzer çalışmalara değinilmektedir. Daha sonra, önerilen algoritma anlatılarak başarı oranı benzer çalışmalarla karşılaştırılmaktadır. Ardından, incelenen yöntemlerin ve önerilen algoritmanın, yeni nesil ARM [5] çekirdekli mikrodenetleyiciler üzerinde çalışma süreleri sunulmaktadır. Elde edilen sonuçlar ise sonuç bölümünde özetlenmektedir.

II. HAREKET ALGILAMA

Gözetim sistemleri genel olarak hareket algılama algoritmasına ihtiyaç duyarlar. Bu işlem görüntüdeki hareketli bölgenin belirlenmesinde kullanılır. Hareket algılama temel olarak, ortamın modellenmesi, hareketin belirlenmesi ve hareket eden nesnelerin sınıflandırılması olmak üzere üç parçadan oluşur [6].

A. Ortamın Modellenmesi

Sabit kameraların kullanıldığı sistemlerin ana problemi arka plan görüntüsünün modellenmesidir [6]. Arka plan görüntüsünün modellenmesindeki ilk adım, başlangıç arka planının oluşturulmasıdır. İlk arka plan görüntüsünü oluşturmak için kameranın algıladığı ortamda hareket yok iken ortamın görüntüsü alınır ve bu görüntü ilk arka plan görüntüsü olarak atanır. Arka plan görüntüsünün modellenmesindeki ikinci adım ise oluşturulan arka planın güncel tutulmasıdır. Arka planın güncel tutulması sayesinde ortamdaki parlaklık değişikliklerinin ve ortama hareketsiz yeni unsurların eklenmesinin hareket olarak algılanmasının önüne geçilmektedir. Arka planı güncelleme amacı ile, görüntü serilerinin zamansal ortalaması, adaptif Gaussian hesabı, piksel işlemleri tabanlı parametre hesaplanması yöntemlerinin başı çektiği bir çok algoritma kullanılmaktadır [6].

B. Hareketin Belirlenmesi

Hareketi belirlemek için kullanılan yöntemler, genel olarak üç temel başlık altında toplanmaktadır [6].

1) *Arka Plan Çıkarımı*: Arka planın sabit olduğu ortamlarda kullanılması uygundur [6]. Bu yöntemde, kameradan alınan görüntü arka plan görüntüsünden piksel piksel çıkartılır. Oluşan fark değerleri, dinamik olarak belirlenen veya daha önceden belirlenmiş sistemin gereksinimlerine uygun bir eşik değerinden büyük ise, piksel hareketli olarak işaretlenir.

2) *Zamansal Değişim*: Zamansal değişim yönteminde, o anki çerçevedeki her piksel daha önceki n sayıda çerçevedeki denk düşen piksellerle karşılaştırılır. Bu karşılaştırma sonucunda bir pikselin hareketli olup olmadığı anlaşılır ve buna göre piksel işaretlenir. Bu çalışma dahilinde incelenmiş Dai'nin [7] ve Wei'nin [8] çalışmaları bu yöntemi kullanmaktadır.

3) *Optik Akış*: Optik akış yönteminde hareket eden cisimlerin akış vektörlerinin karakteristikleri kullanılarak bir görüntü dizisindeki hareketli bölgeler bulunmaktadır.

C. Nesnelerin Sınıflandırılması

Görüntüdeki hareket eden nesnelerin bulunup, bu nesnelerin şekillerine veya hareket biçimlerine göre sınıflandırılmasıdır [6].

III. BENZER ÇALIŞMALAR

Literatür taraması, iki temel kriter göz önüne alınarak gerçekleştirilmiştir. Bu kriterlerden ilki, incelenen çalışmalarda önerilen yöntemlerin kesirli sayı ile işlem yapmamasıdır. Kesirli sayılar ile yapılan işlemler fazladan işlem zamanına ihtiyaç duyduğundan, TCAA düğümlerinde kesirli sayılar kullanılmayarak işlem zamanı optimizasyonu yapılmaktadır. Diğer kriter ise çalışmalarda önerilen yöntemlerin işlem yükünün hafif ve başarımlı değerinin yüksek olmasıdır.

Bu kriterlere uygun görülerek incelenmiş çalışmalardan ilki Dai'nin [7] çalışmasıdır. Bu çalışmada ARM9 çekirdekli gömülü bir görüntü gözetim sistemi kullanılmıştır. Hareket belirleme yöntemi olarak zamansal değişim yöntemi kullanıldığından arka plan modellenmesi yapılmamaktadır. Ardarda gelen dört çerçeveden sonuncusu giriş çerçevesi olarak adlandırılmaktadır. Giriş çerçevesinin kendinden önce gelen çerçevelerle farkları alınmakta ve üç fark çerçevesi oluşturulmaktadır. Ardından, fark çerçevelerindeki piksellerin değerleri bir T eşik değeri ile karşılaştırılmaktadır. Bir pikselin değeri T eşik değerinden küçük ise piksel hareketsiz, büyük ise piksel hareketli olarak işaretlenmektedir. Hareketli piksellerin 1, hareketsiz piksellerin 0 olduğu varsayılarak, fark çerçevelerinin karşılıklı pikselleri mantıksal VEYA işlemine tabi tutulmaktadır. Bu işlem sonucu ortaya çıkan çerçeveye son olarak morfolojik açma ve kapama operasyonları uygulanmakta, bu sayede görüntüden gürültüler temizlenmekte ve görüntüdeki kopukluklar ortadan kaldırılmaktadır. Çalışmada T eşik değerinin, Otsu algoritması [9] kullanılarak elde edildiği belirtilmesine rağmen Otsu algoritmasının hangi çerçeveye uygulanacağı belirtilmemiştir. Bu belirsizlik nedeniyle çalışma gerçekleştirirken, T eşik değerinin, giriş çerçevesine Otsu uygulanarak elde edilmesine karar verilmiştir.

İncelenmiş diğer bir çalışma, Wei'nin [8] ARM tabanlı mikro işlemcileri hedef alarak hazırladığı çalışmasıdır. Bu çalışmada, bir çerçevedeki hareketli bölgeler belirlenirken arka arkaya gelen çerçevelerin farklarından yararlanılan bir yöntem önerilmektedir. Yöntemde kameradan alınan bir çerçeve, kendinden önce gelen çerçeve ile karşılaştırılmakta ve bir fark çerçevesi oluşturulmaktadır. Oluşturulan fark çerçevesindeki pikseller bir T eşik değerinden küçük ise pikseller hareketsiz, büyük veya eşit ise pikseller hareketli olarak işaretlenmektedir. Kullanılan T eşik değeri, sisteme başlangıç anında yüklenen, ortamın hareketsiz olduğu bir çerçeveye Otsu algoritması uygulanması, ardından bulunan eşik değerinin 20'ye bölünmesiyle elde edilmektedir. Son olarak çerçeveye morfolojik açma ve kapama operasyonları uygulanmaktadır.

IV. ÖNERİLEN ALGORİTMA

Bu çalışmada önerilen algoritmanın amacı, kameradan alınan görüntüdeki hareketli bölgeleri belirlemek ve bu bölgelerin, görece işlem yükü daha yüksek olan yüz algılama ve tanıma adımlarına aktarılmasına olanak sağlamaktır. Bu sayede ihtiyaç duyulan toplam çalışma süresinin en aza indirgenmesi hedeflenmiştir.

Tasarladığımız algoritmada hareket belirleme yöntemi olarak arka plan çıkarımı yöntemi kullanılmaktadır. Kameralardan alınan görüntüde hareket belirleme işlemi yapılırken uygulanan ilk adım, gelen çerçeveden arka plan çerçevesinin çıkarılmasıyla, fark çerçevesinin elde edilmesidir (1).

$$D_t(x, y) = |I_t(x, y) - B_t(x, y)| \quad (1)$$

(1) ifadesinde, D_t , I_t ve B_t sırasıyla, t anındaki fark, giriş ve arka plan çerçevelerini ifade etmektedir. Fark çerçevesinin oluşturulması sırasında, piksellerin toplam farkı (2) ve kaç pikselin arka plandan farklı olduğu (3) ifadelerine göre hesaplanmaktadır.

$$tDiff = \begin{cases} tDiff + D_t(x, y) & D_t(x, y) \neq 0 \\ tDiff & \end{cases} \quad (2)$$

(2) ifadesinde, $tDiff$ piksellerin toplam farkını, D_t ise fark çerçevesini ifade etmektedir.

$$count = \begin{cases} count + 1 & D_t(x, y) \neq 0 \\ count & \end{cases} \quad (3)$$

(3) ifadesinde, D_t fark çerçevesini, $count$ ise arka plandan farklı piksellerin sayısını ifade etmektedir. Fark çerçevesi oluşturulurken, belirli bir çerçeve sayısında arka plan çerçevesinden farklı pikseller, arka plan çerçevesinde güncellenmektedir (4). Bir pikselin kaç defa arka plan çerçevesinden farklı olduğu ise, çerçeve ile aynı büyüklükte bir matriste saklanmaktadır (5).

$$B_{t+1}(x, y) = \begin{cases} I_t(x, y) & C_t(x, y) = N \\ B_t(x, y) & \end{cases} \quad (4)$$

$$C_{t+1}(x, y) = \begin{cases} C_t(x, y) + 1 & D_t(x, y) \neq 0 \\ 0 & \end{cases} \quad (5)$$

(4) ifadesinde, B_{t+1} , B_t , I_t ve C_t sırasıyla $t+1$ anında kullanılacak arka plan çerçevesini, t anındaki arka plan çerçevesini, t anındaki girdi çerçevesini ve t anındaki piksellerin kaç defa arka plandan farklı olduğunu tutan matrisi ifade etmektedir. (5) ifadesinde, C_{t+1} , C_t ve D_t sırasıyla, $t+1$ ve t anındaki piksellerin kaç defa arka plandan farklı olduğunu tutan matrisleri ve t anındaki fark çerçevesini ifade etmektedir. Fark çerçevesi oluşturulduktan sonra, piksellerdeki ortalama değişim, piksellerin toplam değişiminin, değişen piksel sayısına bölünmesiyle hesaplanmaktadır (6).

$$mean = totalDiff / count; \quad (6)$$

(6) ifadesinde, $mean$ toplam değişen piksellerin değişim ortalamasını, $totalDiff$ fark çerçevesindeki değerlerin toplamını, $count$ ise kaç pikselin arka plandan farklı olduğunu belirtmektedir.

Ortamda hareket mevcut olmadığında da birçok pikselin değeri arka plandan farklı olabilmektedir. Bunun nedeni, ortamdaki ışık değişimleri ve kameraların iç yapısından kaynaklanan gürültülerdir. Gelen çerçevede hareket olmadığı takdirde fark çerçevesinin histogramı, 0-10 aralığında yoğunlaşmaktadır. Fark çerçevesini oluştururken hesaplanan ortalama fark değeri de bu aralıkta oldukça düşük bir değer almaktadır. Gelen çerçevede hareket mevcut ise fark çerçevesinin histogramında 0-10 aralığında yoğunlaşan gürültü kümesinin yanı sıra gürültü kümesinden ayrı bir hareket kümesi de oluşmaktadır.

Hareketin olduğu durumda fark çerçevesine Otsu algoritması uygulanmasıyla, histogramda gürültü ve hareket piksellerinden meydana gelen iki ayrı küme oluşmaktadır. Ardından hareket kümesinde yer alan pikseller işaretlenmekte ve hareket algılama işlemi tamamlanmaktadır.

Otsu algoritması, hareketsiz fark çerçevesine uygulandığında gürültü kümesini ikiye bölerek, hatalı bir sonuç vermektedir. Bu durumda, görüntüde hareket olmadığını anlaşılabilmesi için, elde edilen Otsu değeri ve ortalama fark değeri kullanılmaktadır. Çerçeve hareket olma ve olmama durumlarını ayırt etmek için, Goyette'nin [10] çalışmasında sunulan veri setleriyle tam sayı değerleri kullanılarak yapılan testler sonucunda bulunan Otsu ve ortalama fark değeri arasındaki oran (7) ifadesinde gösterilmektedir.

$$otsuMeanRatio = \begin{cases} \leq 2 & \text{noMotion} \\ > 2 & \text{Motion} \end{cases} \quad (7)$$

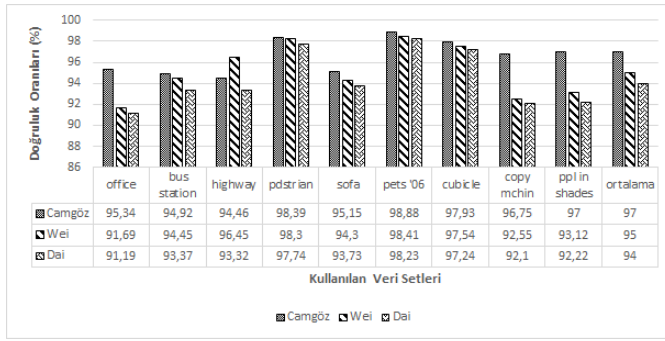
(7) ifadesinde, $otsuMeanRatio$ fark çerçevesine otsu uygulanmasıyla elde edilen eşik değerinin $mean$ ortalama fark değerine bölümüne oranını, $motion$ ve $noMotion$ ise sırasıyla çerçevede hareketin olması ve hareketin olmamasını ifade etmektedir. Herhangi bir çerçeve için oran değerine göre bir sonraki işlem adımına, ya Otsu algoritması uygulanmış fark çerçevesi ya da tüm pikselleri hareketsiz olarak işaretlenmiş çerçeve gönderilir. Bir pikselin hareketli olması 1, hareketsiz olması 0 ile ifade edilir. Otsu algoritması uygulanmış fark çerçevesine, ortalama alan alçak geçirgen filtre uygulanmaktadır. Bu işlem sayesinde görüntüde kalan az sayıdaki gürültüler de temizlenmektedir. Gürültülerden temizlenmiş çerçeve, işlenmek üzere kendinden sonraki adımlara gönderilmektedir.

V. PERFORMANS ANALİZİ

[7], [8] çalışmalarında performans analizi konusunda doğrudan karşılaştırma için kullanılacak değerlerin olmaması ve ortak bir donanım platform kullanılarak bizim yöntemimiz ile karşılaştırmaların gerçekçi olması için belirtilen çalışmalardaki yöntemler gerçekleştirilmiş ve önerdiğimiz algoritmanın başarı ve çalışma süreleri ile karşılaştırılmıştır.

A. Doğruluk Analizi

Doğruluk analizleri, Goyette'nin [10] çalışmasında sunduğu veri setlerinden yararlanılarak, Microsoft Visual Studio 2010 ortamında gerçekleştirilmiştir. Yöntemlerin doğrulukları değerlendirilirken, Goyette'nin çalışmasında sunduğu veri setlerinin işlenmesi sonucu oluşan çıktı çerçeveleri kullanılmaktadır. Bir çerçevenin başarısı, çerçevedeki pikseller ile referans çerçevesindeki ilgili piksellerin karşılaştırılmasıyla elde edilir. Eğer bir piksel çıktı çerçevesinde hareketli, ve referans çerçevesinde de hareketli ise bu piksel doğru, referans çerçevesinde hareketsiz ise bu piksel yanlış olarak işaretlenir. Benzer şekilde eğer bir piksel çıktı çerçevesinde hareketli, ve referans çerçevesinde de hareketsiz ise bu piksel doğru, referans çerçevesinde hareketli ise bu piksel yanlış olarak işaretlenir. Goyette'nin veri setlerin arasından referans çerçeveleri en başarılı şekilde hazırlanmış 9 adet örnek veri seti başarı oranı ölçümünde kullanılmıştır. Dai ve Wei'nin çalışmalarında önerilen yöntemler ve bu çalışmada önerdiğimiz algoritmanın seçilen veri setlerine uygulanmasıyla elde edilen doğruluk değerleri Şekil 1'de görülmektedir.



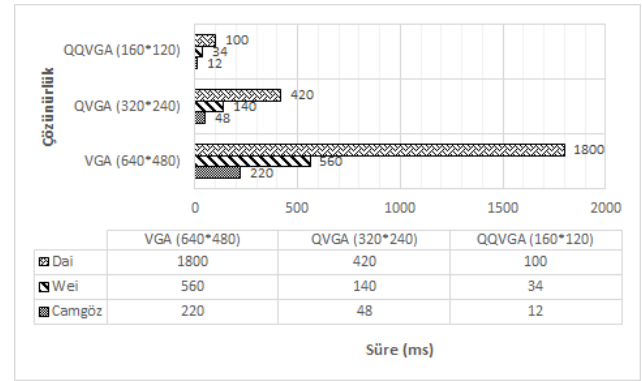
Şekil 1. Tarafımızdan Önerilen ve İncelenen Yöntemlerin Doğruluk Ölçümleri

Yöntemlerin doğruluk analizleri sonucunda, önerdiğimiz algoritmanın %97, Wei'nin yönteminin %95, Dai'nin yönteminin ise %94 oranında ortalama başarıya sahip olduğu tespit edilmiştir. Önerdiğimiz algoritmanın, karşılaştırıldığı yöntemlerden başarılı olmasının sebebi, arka plan çerçevesinin çıkarımı yöntemi ile ortamdaki hareketi belirlemesidir. Wei ve Dai'nin çalışmalarındaki gibi çerçevelerdeki zamansal değişimi değerlendirerek hareket belirleyen yöntemler, çoğu zaman hareket eden nesnelerin sadece sınırlarını tespit edebilmektedirler. Önerdiğimiz algoritma ise, arka plan çıkarımı yöntemini kullanarak, hareket eden nesnenin sadece sınırlarını değil, büyük bir çoğunluğunu tespit etmektedir. Şekil 1'de görülebileceği gibi önerdiğimiz algoritma nerdeyse tüm veri setlerinde, karşılaştırıldığı yöntemlerden daha başarılıdır. Arka planın çok hareketli olması nedeniyle, *highway* veri setinden elde edilen sonuçlar bu başarıya bir istisna oluşturmaktadır.

B. Hız Analizi

Yöntemlerin hız analizleri, incelenen çalışmaların hedeflediği ARM920T çekirdekli S3C2440 mikro-işlemcisine sahip, FriendlyARM firmasının geliştirdiği mini2440 geliştirme ortamı üzerinde gerçekleştirilmiştir [4].

Hız ölçümleri, ilgili geliştirme ortamlarında herhangi bir işletim sistemi kullanılmadan doğrudan kod çalıştırılarak gerçekleştirilmiştir. Ayrıca testler sırasında mikrodenetleyiciler üzerinde bulunan ve kullanılmayan çerçe birimleri kapatılarak, yöntemlerin saf çekirdek performansları ölçülmüştür. Tüm yöntemlerin zaman açısından (-Otime) optimize edilmiş kodları üretilerek test işlemi gerçekleştirilmiştir. ARM920T çekirdeğinin 400MHz frekansında çalıştığı süre ölçüm testlerinin sonuçları Şekil 2'de görülmektedir. Yöntemlerin süre ölçümleri sonucunda, önerdiğimiz algoritmanın, Wei'nin çalışmasından yaklaşık 2.5 kat, Dai'nin çalışmasından ise yaklaşık 8 kat daha hızlı tamamlandığı tespit edilmiştir. Wei ve Dai'nin yöntemlerinde gürültü temizlemek amacıyla morfolojik operasyonlar kullanılmaktadır. İşlem yükü, önerdiğimiz algoritmada aynı amaçla kullanılan ortalama alan alçak geçirgen filtreye göre, ağır olan morfolojik operasyonlar, yöntemler arasındaki hız farkının temel nedenidir. Hız farkının bir diğer nedeni ise Dai'nin yönteminin, önerdiğimiz algoritma ve Wei'nin yöntemine göre belleğe daha çok erişim yapmaya ihtiyaç duymasıdır.



Şekil 2. Önerdiğimiz ve İncelenen Yöntemlerin Ortalama Süre Ölçümleri

VI. SONUÇ

TCAA uygulamalarının sıkça kullanıldığı iç ortam gözetim sistemlerinde, baz istasyonları ile haberleşmeyi en düşük seviyede tutarak ağ trafiğini azaltmak ve istendiğinde baz istasyonu ile erişime olanak sağlamak için ham verinin yerinde işlenerek anlamlı bilgiye dönüştürülmesi kaçınılmazdır. TCAA kullanılarak gerçekleştirilen gözetim sistemlerinde sıkça kullanılan algoritmaların biri hareket belirleme algoritmasıdır. Bu çalışmada, arka planın çok hareketli olmadığı kapalı ortam gözetim sistemlerinde kullanılmak üzere bir hareket belirleme algoritması geliştirilmiş ve mevcut yöntemlerle karşılaştırılmıştır. Gerçekleştirilen testler sonucunda, önerilen algoritmanın, mevcut yöntemlerden hem hız hem de başarımları yönünden üstün olduğu görülmüştür.

KAYNAKÇA

- [1] I. T. Almkawi, M. Guerrero Zapata, J. N. Al-Karaki, and J. Morillo-Pozo, "Wireless multimedia sensor networks: Current trends and future directions," *Sensors*, vol. 10, no. 7, pp. 6662–6717, 2010.
- [2] I. Akyildiz, T. Melodia, and K. Chowdury, "Wireless multimedia sensor networks: A survey," *Wireless Communications, IEEE*, vol. 14, no. 6, pp. 32–39, december 2007.
- [3] R. Zilan, J. Barceló-Ordinas, and B. Tavli, "Image recognition traffic patterns for wireless multimedia sensor networks," in *Wireless Systems and Mobility in Next Generation Internet*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, vol. 5122, pp. 49–59.
- [4] N. Camgoz, R. Ozturk, M. Guvensan, Z. Taysi, and A. Yavuz, "Image processing capabilities of new generation arm cores for wmsns," in *Signal Processing and Communications Applications Conference (SIU), 2012 20th*, april 2012, pp. 1–4.
- [5] ARM. (2013, Feb.). [Online]. Available: <http://www.arm.com/>
- [6] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 3, pp. 334–352, aug. 2004.
- [7] L. Dai, Y. Wang, and J. Wang, "The motion detection method apply to video-surveillance embedded system," in *International Conference on Multimedia Technology (ICMT), 2011*, july 2011, pp. 5162–5165.
- [8] Z. Wei and L. Cai, "Design for motion detection system based on embedded linux," in *International Conference on Multimedia Technology (ICMT), 2010*, oct. 2010, pp. 1–3.
- [9] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, jan. 1979.
- [10] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Change detection.net: A new change detection benchmark dataset," in *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW) 2012*, june 2012, pp. 1–8.