

ANAKENTLERDE KAMU GÜVENLİĞİNE YÖNELİK UYGULAMALAR İÇİN BİR GÜVENLİK KAMERASI AĞI MİMARİSİ

Yrd.Doç.Dr.Harun ARTUNER
Hacettepe Üniv. Bilgisayar Müh.
Hacettepe Üniv. Adli Bilişim
Araş. ve Uyg. Mrk. Ankara,
artuner@hacettepe.edu.tr

Doç.Dr. Cesur BARANSEL
Hacettepe Üniv. Adli Bilişim
Araş. ve Uyg. Mrk.
Danışmanı, Ankara,
cesur@ada.net.tr

Prof.Dr. Turhan MENTEŞ
Hacettepe Üniv. İstatistik
Bölümü, Ankara
mentes@hacettepe.edu.tr

ÖZET

Yıldız biçimli geleneksel güvenlik kamerası ağları tüm dünyada yerlerini hızla sayısal ve yüksek çözünürlüklü IP kameralar içeren ağlara bırakmaktadır. Binlerce kamerayı destekleyebilen bu yeni yapı, daha yüksek kalitede görüntü ve daha yüksek görüntüleme hızları sunmakla birlikte, sıkıştırılması, aktarılması, açılması, incelenmesi, depolanması, dizinlenmesi, sorgulanması, özetlenmesi ve arşivlenmesi gereken verinin boyutlarını da önemli oranda artırmaktadır. Bu ortamda, olağanüstü boyutları nedeniyle söz konusu görsel verinin etkin biçimde yönetilmesi karmaşık bir teknik sorun olup bunun için kapsamlı bir sistem mimarisinin desteği gereklidir. Bu bildiriye, özellikle anakentlerde kamu güvenliğinin sağlanmasına yönelik güvenlik kamerası uygulamalarının geliştirilmesinde kullanılmak üzere tasarlanmış, dağıtım ve ölçeklenebilir bir sistem mimarisi önerisi ele alınmıştır. Önerilen mimari, anakent genelinde hedef izleme ve kameralar arası hedef aktarma gibi görevlerin yanı sıra, görsel verinin içeriğinin otomatik dizinlenmesi ve görsel veri üzerinde içerik-tabanlı arama işlevlerini de destekleyebilecek altyapıya sahiptir.

Anahtar Kelimeler

Geniş ölçekli güvenlik kamerası ağları, dağıtım mimari, video dizinleme, mega-piksel video çözünürlüğü, IP-kamera.

SUMMARY

A Distributed Surveillance Camera Network Architecture for Public Security Applications in Metropolitan Area

Traditional star-topology based surveillance camera networks are being replaced all over the world with IP networks containing high resolution digital cameras. This new structure can support thousands of cameras and offers higher image quality and faster image capture rates which implies huge amounts of visual data to compress, transmit, uncompress, analyze, store, index, search, summarize and archive. In this environment, efficient handling of visual data is not a trivial task due to its enormous volume and thus, there is a need for overarching system architecture. In this paper, we present a highly scalable distributed system architecture specifically targeted at metropolitan area surveillance applications for public security. The proposed architecture has the potential of supporting a large-

scale surveillance network with network-wide tracking and target-handover capabilities and includes provisions for automated annotation of video content and content-based video search.

Keywords

Large-Scale Surveillance Networks, Distributed Architecture, Video Indexing, Mega-Pixel Video Resolution, IP-Camera.

GİRİŞ

PoE (Power-over-Ethernet) özellikli IP kameraların ortaya çıkması ile binlerce kamera içeren güvenlik ağlarının kurulabilmesi teknik ve ekonomik açıdan olanaklı hale gelmiştir. Bu tür ağlarda geleneksel yıldız bağlantılar yerine teknik açıdan daha karmaşık olan IP-tabanlı ağlar kullanılması gereklidir. Ham görsel verinin işlenmek ve depolanmak üzere tek bir merkeze aktarılması yaklaşımı, ağ içerisindeki kamera sayısı arttıkça hızla işlerliğini kaybeder ve işlem yükünün birden fazla merkez arasında dağıtılması etkinlik açısından kaçınılmaz olur. Birden fazla merkezin işbirliğini gerektiren görevler için, işlem ve depolama merkezleri arasında ham ve simgesel görsel veri alış verışı ve paylaşımı sağlanmalıdır.

Kamusal alanda güvenliğin sağlanmasına yönelik akıllı güvenlik kamerası uygulamalarının geliştirilmesi teknik açıdan karmaşık bir süreç olup yapay us, sinyal işleme, görüntü işleme, bilgisayar ağları ve gömülü yazılım geliştirme gibi pek çok değişik teknoloji alanından uzmanlık bilgisi gerektirir. Üretkenliğin artırılabilmesi için, uygulama yazılımcısı bu karmaşıklıktan katmanlı bir yazılım mimarisi ve katmanlar arası arayüzlerle büyük ölçüde yalıtılmalıdır. İlk geliştirilen uygulamalar eldeki teknolojinin zayıflıklarını ve acil olarak iyileştirilmesi gereken süreçleri hızla ortaya çıkarmıştır. Buna bağlı olarak son on yılda özellikle veri iletişimi ve veri sıkıştırma standartları başta olmak üzere pek çok alt düzey işlevle ilgili önemli gelişmeler sağlanmıştır. Daha önceleri yalnızca görüntü kaydetme ve izleme amacıyla kullanılan veri sıkıştırma yöntemleri (örneğin MPEG-2) IP ağları üzerinden veri aktarımına uygun, iletişim hatalarına karşı daha dayanıklı ve daha yüksek sıkıştırma oranları sağlayacak biçimde uyarlanmıştır (örneğin H.264). Benzer biçimde, TCP-tabanlı iletişim protokollerinden daha etkin olduğu görülen UDP-tabanlı protokollerin (örneğin RTP/RTCP (Real-time

Transport Protocol/Real-time Transport Control Protocol)) kullanımı da giderek yaygınlık kazanmaktadır.

Görsel verinin işlenmesi ve saklanması ile ilgili olarak, kapsamlı bir sistem mimarisinin gerekliliği ortaya çıkmış ve bu konu birkaçına izleyen bölümde kısaca değineceğimiz çeşitli bildirilerde ele alınmıştır. Ancak getirilen çözüm önerileri genellikle konunun yazılım geliştirme yönüne değinmemekte ve mega-piksel görüntü verisinin gerçek-zamanlı ya da gerçek-zamanlıya yakın hızda işlenmesinin doğuracağı sonuçları ele almakta yetersiz kalmaktadır. Bu bildiride, akıllı güvenlik kamerası uygulamalarının anakent ölçeğinde kullanılabilecek nitelikte geliştirilebilmesi amacıyla tasarlanmış yeni bir sistem mimarisi öneri yer almaktadır.

İLGİLİ ÇALIŞMALAR

Desurmont ve diğerleri CCTV (Closed-circuit television) de dâhil olmak üzere değişik kamera standartlarını destekleyebilen bir kamera platformu geliştirmiştir [4]. Önerilen yapıda, yüksek düzeyli görüntü işleme birimleri uç-kullanıcıya yönelik uygulamalara seçimli olarak bağlanabilmektedir.

CLOVIS adı verilen ve Enfciaud ve diğerleri tarafından tasarlanmış sistem mimarisi, video çözümlene uygulamalarının tek bir işlemci üzerinde ya da dağıtımli olarak çalışabilecek biçimde geliştirilebilmesine olanak sağlamaktadır [6]. CLOVIS, esas olarak XML belirtimleri, RPC-benzeri bir iletişim mekanizması ve aktarma katmanında SOAP protokolü kullanan olay-tabanlı bir mimaridir.

Detmold ve diğerleri güvenlik kamerası ağları için bir orta-katman yazılımı önermektedir [5]. Önerilen orta katman yazılımı, işlemler için karatahta mimarisi, iletişim için ise hizmet-odaklı mimari kullanmaktadır. Karatahta, dikey ve yatay olarak aralarında etkileşim bulunan düzeylere bölünmüştür. Dikey düzlemde, karatahtanın farklı düzeyleri farklı işlem birimlerine atanabilecek biçimde öngörülmüştür. Değişik izleme bölgeleri (kamera grup-ları) ise karatahtayı yatay olarak bölmektedir. Sistemin tasarımcıları, olası bir tehdiye insan gözlemcinin tepki süresinin 5 dakika civarında olması nedeniyle, bu tür uygulamaların zaman-kritik sistemler kapsamında ele alınmamasını savunmaktadır.

Wijnhoven ve diğerleri sistem mimarisinin gereklerini belirlemek için öncelikle değişik uygulamaları incelemekle işe başlamıştır [13]. Belirlenen dokuz sistem gereğinden yola çıkarak önerdikleri mimari özellikle işlem gücünün ölçeklenebilirliği ve birimselliği üzerinde durmaktadır. Benim-senen temel yaklaşım, her kamera için ayrı bir etki alanı oluşturulması ve bu alan içerisinde birden fazla VCA (Video Content Analysis) birimi, bir adet kameradan görüntü toplama birimi, etki alanı içinde zaman-uyumlu iletişim için bir yöneltici ve diğer etki alanları ile zaman-uyumsuz iletişim için giriş ve çıkış birimleri kullanılması esasına dayanmaktadır.

Fonseca ve diğerleri güvenlik kamerası uygulamalarının yalnızca optik-akış tekniği kullanılarak gerçekleştirilebileceğini savunmaktadır [7]. Böylece her biri değişik algoritmalar ve teknik-ler kullanan farklı işlem birimlerinden kaynaklanan hesaplama yükü ve artıklık ortadan kalkmış olacaktır.

DiVA, Miguel ve diğerleri tarafından önerilen bir istemci-sunucu mimarisidir [11]. Söz konusu mimaride, biri kameralar diğeri kişisel bilgisayarlar için olmak üzere birbiriyle iletişimi olan fakat ayrı tutulmuş iki IP ağı yer almaktadır. Sistemin mantıksal tasarımı, adlarından anlaşılabilen işlevler yüklenmiş, veri toplama katmanı, iletişim katmanı, işlem birimleri katmanı ve veri yönetimi katmanı adı verilen dört katmandan oluşmaktadır. Bildiride yer alan ilk-örnek tek bir kişisel bilgisayar içermektedir.

Çin'in DongGuan kentinde gerçekleştirilen E-Touch projesi 2000 güvenlik kamerası ve 600 TB görüntü depolama kapasitesi olan bir güvenlik kamerası sistemidir [3]. Söz konusu sistem, iletişim ağı katmanı, hizmet ağı katmanı ve uygulama katmanı adı verilen üç katman içermektedir. Bir hizmet istemi alındığında, hizmet ağı katmanı hizmet bütünleştirme için gerekli birimleri belirleyerek bu birimler arasında belirli ölçüde hizmet kalitesini (QoS) koruyacak bağlantılardan oluşturduğu bir iletişim kanalı kurar. Bildiride sunulan benzetim sonuçları, 20–80 birim arasında her biri 128–512 Kbps bant genişliği gerektiren hizmet istemleri ve toplam 512 Kbps–10 Mbps hizmet ağı kanal genişliğine sahip bir sistem göz önüne alınarak hesaplanmıştır.

Akıllı ve dağıtımli güvenlik kamerası uygulamalarının genel bir incelemesi Valera ve Velastin tarafından yapılmıştır [12]. Aghajan and Cavallaro, çoklu-kamera sistemlerinin değişik yönlerini diğer araştırmacıların da katkısıyla bir kitapta ele almaktadır [1].

ÖNERİLEN MİMARİ

Yukarıda değinilen ya da benzer çalışmalar incelendiğinde, kamera bunu sağlayabilecek nitelikte olsa bile yüksek çözünürlüklü görüntünün aktarılmasından işlenmesinden ve saklanmasından elden geldiğince kaçınma yönünde genel bir eğilim göze çarpmaktadır. Örneğin, bir çalışmada D1 çözünürlükte (1280x720) EPTZ kameradan gelen iki ayrı duraksız veri akımı aynen bu yaklaşımla ele alınmıştır [2]. Söz konusu sistem düşük çözünürlüklü duraksız veriyi baseball oyuncularını bulmak ve geri plan görüntüsü oluşturmak için kullanır. Daha sonra yalnızca oyuncularını içeren bölümler yüksek çözünürlüklü duraksız veriden kesilerek düşük çözünürlüklü görüntünün üstüne yapıstırılır ve elde edilen görüntü izleyicinin ekranına yansıtılır. Benzer yaklaşım geniş açılı bir kameranın hedef nesnelerin ayrıntılı görüntülerini çekmek üzere bir PTZ kamerayı yönlendirdiği başka çalışmalarda da kullanılmıştır [10],[14]. Bu arada benim-senen temel yaklaşım düşük çözünürlüklü video görüntüsünü eksiksiz kaydetmek ve yüksek çözünürlüklü bölgesel görüntüleri şu ya da bu biçimde

düşük çözünürlüklü video kaydıyla birlikte kullanılmaktadır. Her durumda, gözlenen alanın yüksek çözünürlükte bir kaydının sonradan incelenmek üzere saklanması feragat edilmektedir. Çok sayıda izleyiciye kısıtlı bir bant genişliği üzerinden hizmet üretmek üzere tasarlanan ve görüntüdeki çözünürlük kaybının önemli olmadığı uygulamalar dışında, söz konusu yaklaşımın güvenlik uygulamaları açısından sakıncaları aşağıda sıralanmıştır:

a) Güvenlik uygulamalarında takip edilecek nesnelere önceden belirlenmesi olanaklı değildir. Örneğin bir hava alanında güvenlik kamerasının görüş alanına giren kişilerden herhangi birisi olağanüstü bir durum söz konusu yaşanmadıkça diğerinden daha özel değildir. Benzer biçimde daha sonraki bir soruşturma için hangi nesne ya da kişinin önemli olacağı da önceden bilinemez. Bu nedenle, yüksek çözünürlüklü görüntünün sürecin en başında gözden çıkarılması yaklaşımı sakıncalı bir yaklaşımdır. Bu görüşü destekleyen haberler basında zaman zaman yer almaktadır. Örneğin, 2006'da Cologne, Almanya'da metroda bomba patlatmaya çalışan bir şüpheli güvenlik kameralarına yakalanmış, ancak çekilen görüntülerin Alman televizyonlarında hemen yayınlamasına rağmen bu görüntülerin bulanık olması nedeniyle teröristin yakalanması ya da kimliğinin saptanması olanaklı olmamıştır [9]. Bu ve benzeri olayların verdiği mesaj oldukça açıktır; güvenli kamerası görüntüleri bir çeşit sigorta olarak değerlendirilmelidir. Her tür sigortada olduğu gibi sigortaya gerek duyulacak duruma düşmemek en iyisi olmakla birlikte, gereksinim ortaya çıktığında beklenen yararın sağlanabilmesi için sistem yeterince yüksek çözünürlükte görüntüyü sağlayabilmelidir.

b) Yapılan her sorguda video kayıtlarının baştan sona kare kare incelenmesi etkin bir yöntem değildir. Ayrıca "bu kişi son iki hafta içinde bu civarda kaç kez görüldü?" türü sorguların bu yöntemle yanıtlanması son derece zahmetlidir. Hızlı ve otomatik tarama için görüntü kayıtlarının, tercihen insan katkısı gereksiz ve ek açıklamalar da içerecek biçimde dizinlenmesi gerekir. Video kayıtlarına otomatik açıklama oluşturmak için genel olarak kullanılan yöntemde önce görüntü kaydının kesimlemesi yapılır. Sonra önceden belirlenmiş nesne türlerini içeren kesimler ayıklanarak her nesne için bir nitelik vektörü ya da bir matematik model hesaplanır ve hesaplanan model ya da nitelik vektörü görüntü kaydına işaret edecek biçimde saklanır. Sorgu aşamasında, sisteme girilen görüntüden elde edilen sorgu vektörü sistemde depolanmış olan vektörlerle uygun bir benzerlik ölçütüne göre karşılaştırılır. Belirli bir eşik değerin üstünde benzerlik gösteren vektörlerin işaret ettiği görüntüler son değerlendirme için insan kullanıcıya sunulur. Bu yöntemde başarıyı etkileyen en önemli unsur, hesaplanan model ya da nitelik vektörünün

ayırım gücü olup, bu da doğrudan modelin hesaplanmasında kullanılan görüntünün kalitesine bağlıdır. Sonuç olarak, video kayıtlarının dizinlenmesinde yüksek çözünürlüklü görüntü kullanımının başarımı artırdığı açıktır.

c) Yeni geliştirilen yüksek çözünürlüklü kameralar oldukça geniş alanları yeterli ayrıntıda kapsayabilmektedir. Örneğin 16 mega-piksel çözünürlükte bir kamera, tek bir resim karesi içinde 50 adet VGA kameraya eşdeğer görüntü verisi sağlamaktadır (Şekil 1). Kurulum ve bakım maliyetlerinin çok daha düşük olmasının dışında, tek kameradan alınan verinin incelenmesi ve yönetilmesi 50 ayrı kameradan toplanacak veriye kıyasla daha kolaydır.

Diğer taraftan, kameraların çözünürlüğü arttıkça güvenlik kamerası sistemlerinin desteklediği çözünürlük düzeyini de benzer oranda otomatik olarak artırmak olanaklı değildir. İletişim ağlarına ve görüntü depolama birimlerine gelecek büyük ek yükün dışında sadece daha çok kaynak kullanılarak çözülemeyecek başka teknik sorunlar da söz konusudur. Örneğin, piksel bazında geri plan görüntüsü modeli oluşturulması gibi bazı temel tekniklerin gerçek zamanlı kullanımı belirli bir çözünürlük düzeyinin ötesinde olanaklı değildir. Bu nedenle, geniş ölçekli sistemlerde mega-piksel çözünürlük kullanımının sonuçlarının önceden dikkatle incelenmesi ve sistem kaynakları üzerinde genellikle birbirine ters düşen istemler içeren bazı parametrelerin ölçeklenebilir, verimli ve güvenli bir mimari oluşturacak biçimde karşılıklı olarak dengelenmesi gerekir. İzleyen alt bölümde öncelikle bu tasarım parametreleri ele alınacak ve daha sonra önerilen mimari açıklanacaktır.

Tasarım Parametreleri

Bir görüntü karesinin akıllı güvenlik kamerası sistemi içindeki yaşam döngüsü ana hatlarıyla aşağıdaki aşamalardan oluşur; a) resim karesi kamera tarafından oluşturulur, b) görüntüleneceği ve/veya işleneceği noktaya iletilir, c) kısa dönemli erişim için sayısal olarak kaydedilir, d) uzun dönemli erişim için arşivlenir ve e) sistemden silinir. Yaşam döngüsünün her aşaması çeşitli sistem görevleri ile ilişkilidir ve bir sistemin mimarisi bu görevlerin nerede ve nasıl yerine getirildiğinden yola çıkılarak tanımlanabilir.

Sistem ölçeklenebilirliğinin en yüksek düzeye taşınabilmesi için, bir resim karesinin yaşam döngüsünün büyük bölümünü kamera içerisinde tamamlaması gerekir. Eğer kamera gerekli işlem gücüne, depolama kapasitesine ve iletişim yeteneğine sahipse ve yeteri derecede akıllı uygulamalarla donatılmışsa, resim karesinin insan kullanıcının izlemek istediği durumlar dışında kamerayı terk etmesi gerekmez. Bu durumda, kamera yalnızca önemli olayları rapor eder, soruşturma sorgularına yanıt verir ve ağ genelindeki hedef izleme ve saptama süreçlerinde diğer kameralarla işbirliği



Şekil 1. 16 Mega-piksel çözünürlükte resim karesi; sayfaya sığması için küçültülmüş olup gerçek ayrıntı düzeyinden örnekler beyaz dikdörtgenler içerisinde gösterilmiştir (www.avigilon.com, 2010)[15].

yapar. Böylece, gerekli bant genişliği, merkezi işlem gücü ve merkezi depolama kapasitelerinde önemli tasarruflar sağlanabilir. Tüm bunlar, “akıllı-kamera” ağları olarak adlandırılan yapının dayandığı temel özelliklerdir [1]. Ancak, henüz kameraların şu anda insan işletmenler tarafından yürütülen görüntü değerlendirme ve karar verme görevlerini kendi başlarına yürütecek kadar akıllı olmaması nedeniyle, işlem ve depolama merkezlerinin yakın gelecekte ortadan kalkması pek olası görünmemektedir. Ayrıca olay sonrası soruşturmalar geriye dönük olarak ve depolanmış görüntü kayıtları incelenerek yapılmaktadır.

Kamera görüntüsünün izlenmek, işlenmek ve depolanmak üzere kamera dışına taşınması gereksinimi bazı sistem parametrelerinin dikkatle incelenmesini ve bir bütün olarak eniyilenmesini gerektirir. Genel olarak, bu parametreler sistem kaynakları üzerinde birbirleriyle çelişen istemlere karşılık gelir ve bu nedenle sistemin temel özelliklerine ilişkin bazı tercihlerin en baştan yapılması kaçınılmazdır. Aşağıda önce bu parametreler açıklanacak ve daha sonra mimari tasarıma esas olacak tercihler yapılacaktır.

- *Görüntü Çözünürlüğü*, sistemin üreteceği ve yönetmesi gereken görsel veri boyutunu belirlediğinden sistemin en önemli parametresidir. Genel olarak, çözünürlük ne kadar yüksekse o kadar fazla kaynak kullanılması gerekir.
- *Görüntüleme Hızı ve Toplam Kamera Sayısı* sistemin ürettiği toplam görsel veri boyutu hesaplanırken çarpan değerler olarak kullanılır.

- Bazı IP kameralar değişik çözünürlük ve görüntüleme hızına sahip birden fazla duraksız veri akımını eşzamanlı olarak üretebilmektedir. *Eşzamanlı Akım Sayısı* sistemin ürettiği toplam görsel veri boyutu hesaplanırken çarpan değer olarak kullanılır.
- Sistem içerisinde aktarılabilecek veri boyutu sistemin ürettiği görsel veri boyutu ile ilişkili olmasına karşın, bu değere eşit değildir. Eğer kameralar ürettikleri her veriyi bir başka noktaya aktarmıyorsa daha küçük, eğer birden fazla noktaya iletiyorsa daha büyük boyutlar söz konusu olacaktır. Gerekli *Bant Genişliği* sistemde aktarılabilecek veri boyutuna göre hesaplanır ve bu hesaplamada *Sıkıştırma Katsayısı* bölen olarak kullanılır.
- *Depolama Kapasitesi* doğrudan sistemde üretilen veri boyutu ve bu verinin öngörülen saklama süresi ile ilgilidir. Depolama kapasitesi hesaplanırken *Sıkıştırma Katsayısı* bölen olarak kullanılır. Ayrıca, daha hızlı arama ve eşleştirme amacıyla, depolanan görüntünün ek açıklama ve dizinlerle desteklenmesi durumunda gerekli depolama kapasitesinin artacağına dikkat edilmelidir.
- *İşlem Gücü* görüntü çözünürlüğü, sıkıştırma yöntemi ve kullanılacak görüntü çözümleme uygulamalarının karmaşıklığı ile doğrudan ilgilidir. Sistemin işlem gücü daha hızlı işleyiciler, kodlama, kod çözme ve koşturulan özel donanım birimleri ve daha hızlı depolama birimleri kullanılarak artırılabilir.
- *Gecikme Süresi* sistemin yanıt süresini tanımlar ve genel olarak iletişim hatlarının hızına, kullanılan bant

genişliğine, işlem gücüne ve görüntü çözünürlüğüne bağlıdır.

Önerilen mimari tasarımın başlangıç noktasını aşağıdaki iki temel tercih oluşturmaktadır;

- Örneksele güvenlik kameralarının yakında büyük ölçüde kullanımdan kalkacağı yönündeki öngörümüz doğrultusunda önerilen mimari yalnızca sayısal IP kamera ağları için tasarlanmıştır.
- Önerilen sistem mega-piksel görüntü çözünürlüğü için tasarlanmıştır. Bunun altında bir çözünürlüğe yönelmek, yüksek çözünürlüklü görüntünün sağladığı üstünlüklerden daha en başta vazgeçmek olacaktır. Bir güvenlik kamerası sisteminin ancak sağladığı görüntünün kalitesi oranında yararlı olduğu açıktır.

Yukarıdaki iki tercih tasarımın temelini oluşturmaktadır. İletişim ağının türü ve görüntü çözünürlüğü belirlendikten sonra diğer tasarım kararları hızla alınabilir. Yüksek çözünürlüklü görüntünün IP ağı üzerinden verimli biçimde aktarılması ve depolanması için eldeki en etkin sıkıştırma standardının, bir başka deyişle H.264'ün kullanılması gerekir. Bu noktadan hareketle akım başına gerekli bant genişliği hemen hesaplanabilir. Gerekli disk kapasitesinin hesaplanması için toplam kamera sayısı, kamera başına akım sayısı ve görüntü kayıtlarının saklanma süresi belirlenmelidir. Önerilen mimarinin ayrıntıları izleyen bölümde açıklanmıştır.

Önerilen Mimarinin Temel Yapısı

Yüksek çözünürlüklü görsel verinin özellikleri nedeniyle eldeki sistem kaynaklarının çok verimli bir biçimde yönetilmesi zorunludur. Video çözümleme uygulamaları işlem-yoğun uygulamalar olup, orta karmaşıklıkta bir uygulama bile yüksek çözünürlük söz konusu olduğunda standart bir kişisel bilgisayarın kaynaklarını tek başına tüketebilir (bu nedenle ticari uygulama paketlerinin çoğunda D1'den yüksek çözünürlükler için video çözümleme işlemi yapılamamaktadır). Buna ek olarak, görsel duraksız veri akımlarının ağ trafiği özellikleri geleneksel veri işlem uygulamalarından çok farklı özelliklere sahiptir. Görsel veri akımı, uzun süreli, yoğunlukla tek yönlü ve kararlı bir trafik üretirken geleneksel uygulamalar, kısa süreli, ani sıçramalar gösteren ve uç-kullanıcının değerlendirme molalarıyla kesilen bir trafige sahiptir. Bu nedenle, önerilen mimaride iletişim kanallarının ve işlem kaynaklarının yönetiminde kesen kaynak atama yöntemi kullanılmıştır. Mimari aşağıda açıklanan dört katman biçiminde düzenlenmiştir.

Altyapı Katmanı (AK)

Genel olarak bu katman, kameralar, bilgisayarlar, disk sürücüler, kodlayıcılar, kod çözücüler gibi IP tabanlı güvenlik kamerası ağının kurulması için gereken tüm yazılım ve donanım bileşenlerinin tanımlandığı katmandır. Kullanılacak standartlar, iletişim ağının fiziksel ilingesi, kabloların türü ve uzunluğu, yöneticilerin sayısı ve özellikleri, depolama sunucularının kapasiteleri, kamera konumları ve güç kaynakları bu

katmanda belirlenir. Önerilen mimaride, bu katmanda tanımlanan kaynaklar izleyen kesimde açıklandığı üzere KYK tarafından yönetilir.

Kaynak Yönetim Katmanı (KYK)

KYK, sistem içerisindeki tüm kaynakları yönetir. Bu nedenle, sistemin ayrıntılı ilingesi dâhil eldeki tüm kaynaklar hakkında tam bilgiye sahiptir. Bir kameradan görüntü almak isteyen herkes (insan ya da bilgisayar) bu isteminin KYK tarafından değerlendirilmesini, geçerlenmesini ve gerekli bağlantının kurulmasını beklemek zorundadır. Uzak sistemlerden yapılan bağlantı istemleri de KYK tarafından yönetilmekte ve böylece sisteme güvenli bir tek giriş noktası tanımlanmış olmaktadır. Yöneticiler dışarıdan yapılacak doğrudan bağlantı isteklerini hemen reddedecek biçimde ayarlanmalıdır. KYK'nın ana görevleri aşağıda sıralanmıştır;

- Kullanıcıların yönetimi.
- Ağ yönetimi ve izleme.
- [İzleme istasyonu↔kamera], [izleme istasyonu↔depolama sunucusu], [işlem birimi↔kamera] ve [işlem birimi↔depolama sunucusu] bağlantılarını sağlamak yoluyla erişim denetimi.
- Kamera yönetimi.

Mevcut durumda, bu katmandaki görevlerin yerine getirilebilmesi için marka ve model bağımlı birden fazla yönetim yazılımının kurulması gerekmektedir. Eğer ağda birden fazla marka kamera kullanılırsa, bu kameraların kurumlarını yapan ve görüntüyü depolama sunucusuna yönlendiren işletmen bu işler için ayrı programlar kullanmak zorunda kalabilir. Bu kadarı sistemin statik yapısının belirlenmesi için yeterlidir. Ancak uygulama yordamlarından gelecek bağlantı istemlerinin işletmen eliyle yönetilmesi olası olmadığından uygun bir API ya dışarıdan edinilmeli ya da kurum içerisinde özel olarak geliştirilmelidir.

Gözetim API'si Katmanı (GAK)

Bu katman güvenlik kamerası uygulamalarının geliştirilmesi, dağıtılması ve çalıştırılması için gereken yapısal bileşenleri içerir ve aşağıdaki ana hizmetleri sunar;

- Görevlerin işlem birimlerine atanması ve bu görevler arasında hizmet istemlerinin ve sonuçlarının dağıtılması için gerekli eşgüdümü sağlayan altyapı.
- Kameralara erişim için kullanılacak üst düzeyli birörnek arayüz.

Üst düzeyli birörnek kamera arayüzü, marka bağımlı kamera API'lerini sarmalayan bir yapıda tasarlanmıştır. PTZ ve sabit kameralar (optik yakınlaştırmalı da olabilir) için ayrı arayüzler öngörülmüştür. Söz konusu arayüzlerin amacı kamera erişiminde kullanılacak ortak bir komut kümesinin (en azından kamera üreticileri bir standart üzerinde anlaşana kadar) oluşturulmasıdır. Bu arayüzlerin ayrıntıları bildirinin kapsamı dışındadır.

İşlem yönetimi altyapısı, bir ana yordamın fiziksel olarak dağıtılmış işlemciler üzerinde eşzamanlı görevler başlatılmasına ve bunlar arasında gerekli eşgüdümü sağlayabilmesine olanak verir. Konuya yönelik temel yaklaşım izleyen bölümde yer açıklanmaktadır.

Video Uygulamaları Katmanı (VUK)

Bu katmanda yalnızca GAK tarafından sunulan hizmetleri kullanacak biçimde geliştirilmiş güvenlik kamerası uygulamaları yer alır.

ÖRNEK UYGULAMALAR

Bu bölümde üç uygulamanın kavram düzey kodları ve bu uygulamaların sistem içerisinde nasıl dağıtıldığı ele alınacaktır. Sunulan kod parçalarında GAK tarafından sunulan hizmetler SAL (*Surveillance API Layer*) öneki ile belirtilmiştir.

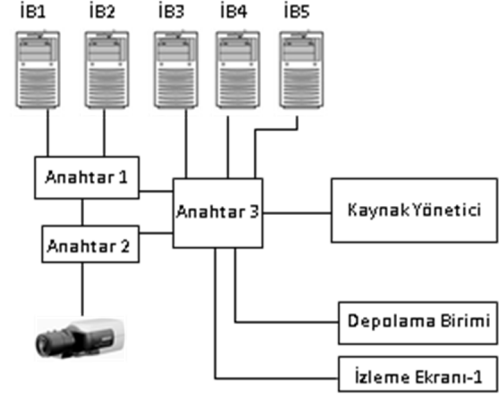
Bu uygulamaların ele alınmasındaki esas amaç, güvenlik kamerası uygulamalarının geliştirilmesi ile ilgili olarak benimsediğimiz yaklaşımın temel özelliklerinin açıklamaktır. Örnekler incelenirken, yukarıda tanımlanan katmanlı yapının uygulama yazılımcısını ağ ve çoklu-kullanım programlarının inceliklerinden nasıl yalıtıldığına ve işlem-yoğun görevlerin yerine getirilmesi için sistem içerisindeki işlem kapasitesinin çok yalın bir arayüzle kullanılabilmesine dikkat edilmelidir. Ayrıca, çözünürlük ne kadar yüksek olursa olsun, dizinleme işleminin gerçek ya da gerçek zamana yakın sürelerde tamamlanması gerektiği ve sistemin bu amaçla değişik algoritmalarla deney yapılmasına olanak sağlayacak bir ortamı desteklemesi gerektiği görüşündeyiz. Sunulan örnekler sistemin bu niteliklerini vurgulamak üzere özellikle seçilmiştir.

Video Kayıtlarının Resimlerle Dizinlenmesi

Uygulamanın, bir IP kameradan alınan 5 mega-piksel çözünürlükte duraksız veride insan yüzlerini saptaması, bu yüzlerin küçük resimlerini oluşturması, uygun belirtilerle birlikte ve video kaydı ile de ilişkilendirerek saklaması istenmektedir. Bu uygulama, Hampapur ve diğerleri tarafından da betimlenen “Yüz Katalogu” uygulamasının ana bileşenlerinden birisidir [8].

Uygulamanın kullanılacağı sistemin yapısı Şekil 2’de verilmiştir. İşlem birimleri (İB) standart kişisel bilgisayarlardır. IP kamera ise değişik görüntüleme hızında ve çözünürlükte 8 duraksız veri akımını eşzamanlı olarak destekleyebilmektedir. En yüksek çözünürlükteki H.264 akımının kesintisiz olarak depolama birimine kaydedildiği ve istendiğinde İzleme Ekranı 1’de izlenebileceği varsayılmaktadır. Üç anahtarın varlığı ise artık olarak düzenlenmiş bir iletişim ağına işaret etmektedir.

Bir tek kişisel bilgisayarın tanımlanan tüm görevleri tek başına gerçek zamanlı olarak yerine getiremeyeceği açık olduğundan uygulamanın 5 işlem birimine dağıtılması öngörülmüştür. Ana yordamı işleten İB, tam çözünürlükte kareleri saniyede 9 kare hızında almaktadır. Her kare küçük miktarda kesilecek biçimde 4 parçaya bölünerek her bir parçanın işlenmesi



Şekil 2: Örnek Uygulama için Sistem Yapılanışı

görevi ayrı bir İB’ye verilir. Bu İB’lerden her biri görev bölgeleri ile kısıtlı olmak üzere kameradan kendi duraksız veri akımlarını başlatırlar. Bu veri akımlarından her biri 1.25 mega-piksel görüntü verisini saniyede 24 kare hızında taşıyabilir. Her İB kendi görev bölgesindeki yüzleri bulur, bunları içeren en küçük dikdörtgeni hesaplayarak gerekli bilgileri oluşturur ve ana yordama gönderir. Ana yordam gelen bilgilere göre küçük resimleri keserek gerekli dizinlemeyi yapar. Ana yordamın yüz bulma işlemi yapmamakla birlikte, bir bölgeden diğerine geçen yüzler, farklı görüntüleme hızlarından kaynaklanan sorunlar, diğer yordamlardan gelen bilgiler arasındaki zaman farklarının ele alınması gibi pek çok sorunu çözmek zorunda olduğuna dikkat edilmelidir.

Ana yordamın İB5 üzerinde çalıştığı kavram düzeyi kodu Şekil 3’de verilmiştir. Ana yordam öncelikle kullanılacak İB grubunu oluşturmakla işe başlar. İB’ler de diğer tüm kaynaklar gibi KYK katmanında kayıtlı olup her birinin sistem genelinde biricik bir kimliği vardır. Ana yordam bu kimlikleri kullanarak, bir İB grubu oluşturur ve kendisini de bu grubun ana yordamı olarak belirler. Tasarımda, her grubun tek bir ana yordamı olması ve tüm yordamların yalnızca bu ana yordamla iletişim kurabilmeleri öngörülmüştür. Diğer İB’ler arasında iletişim kurmak olanaklı değildir. Ana yordam oluşturduğu gruba dinamik olarak yeni görev ya da İB ekleyebilir ya da çıkarabilir.

Sistemde var olan her güvenlik kamerası uygulaması KYK’da kayıtlı olup her İB’ye bir kopyası yüklenmiştir. Ana yordam, işletilmek üzere *FaceMBR* uygulamasını seçerek gerekli giriş-çıkış parametrelerini *TaskDescriptor* adı verilen ve aslında bir tablo olan veri yapısına yazar. *FaceMBR* uygulamasının, birisi kullanılacak kameranın kimliği diğeri işlenecek bölgenin koordinatları olmak üzere iki giriş parametresi olup sonuç olarak bir dikdörtgen koordinatları listesi döndürür. Sonuç olarak, uygulamada yer alan görev tanımları arasındaki tek fark işlenecek bölgenin koordinatlarıdır.

Daha sonra ana yordam, görev-İB eşleştirmelerine ve kurulacak iletişim kanallarına ilişkin bilgilerin tutulduğu *SAL_CommStructure* adlı veri yapısını yaratır. *SAL_AssignTasksToPUG* komutu tanımlanmış görevleri İB'lere otomatik olarak dağıtır (görevleri İB'lere seçerek atayabilmek ya da bir İB'ye birden fazla görev yükleyebilmek için *SAL_MapTasksToPU* komutu kullanılmalıdır). *SAL_ActivateTasks* komutu ise iletişim bağlantılarını kurar, görevleri başlatır ve durum kodunu geri döndürür. İletişim bağlantıları kurulurken, gerekli eniyilemeleri yapmak görevi KYK tarafından yerine getirilir.

```

BuildFaceIndex {
    hPUG = new (SAL_ProcessingUnitGroup);
    hPUG.SetMasterProcessingUnit(PU5);
    hPUG.AddProcessingUnit(PU1);
    hPUG.AddProcessingUnit(PU2);
    hPUG.AddProcessingUnit(PU3);
    hPUG.AddProcessingUnit(PU4);

    hTD = new (SAL_TaskDescriptor);

    taskID = SAL_GetTaskIdByName("FaceMBR");
    hTD.AddTask(taskID,hTP);
    hTD.AddTaskParam(taskID,"Camera",CameraId);
    hTD.AddTaskParam(taskID,"CropCoords",Q1);
    hTD.AddTaskParam(taskID,"MBRQ1");

    taskID = SAL_GetTaskIdByName("FaceMBR");
    hTD.AddTask(taskID,hTP);
    hTD.AddTaskParam(taskID,"Camera",CameraId);
    hTD.AddTaskParam(taskID,"CropCoords",Q2);
    hTD.AddTaskParam(taskID,"MBRQ2");

    taskID = SAL_GetTaskIdByName("FaceMBR");
    hTD.AddTask(taskID,hTP);
    hTD.AddTaskParam(taskID,"Camera",CameraId);
    hTD.AddTaskParam(taskID,"CropCoords",Q3);
    hTD.AddTaskParam(taskID,"MBRQ3");

    taskID = SAL_GetTaskIdByName("FaceMBR");
    hTD.AddTask(taskID,hTP);
    hTD.AddTaskParam(taskID,"Camera",CameraId);
    hTD.AddTaskParam(taskID,"CropCoords",Q4);
    hTD.AddTaskParam(taskID,"MBRQ4");

    hComm = new (SAL_CommStructure);
    SAL_AssignTasksToPUG(hPUG, hTD, hComm);
    Status = SAL_ActivateTasks(hComm);

    hCam = SAL_FindCam (CameraId);
    hStream = SAL_OpenStream (hCam,
                             fullResolution,
                             fullRate);
    hBuf = new (SAL_ImageBuffer);

    while (not done) {
        hBuf.GetNextFrame(hStream);
        SVC_GatherResults(hComm, TimeOutValue);
        BuildIndex(hComm);
    }
    SAL_destroy(hComm);
}

```

Şekil 3: Video Dinleme Uygulamasının Kavram Düzeyi Kodu

Ana yordam, kameradan kendi duraksız veri akımını başlatarak ilk karesini alır ve *SVC_GatherResults* komutunu kullanarak diğer İB'lerden gelecek dikkörtgen listelerini beklemeye başlar. Bu komut belirli bir süre sonunda İB1, İB2, İB3 ve İB4' den gelen dikkörtgen listelerini sırasıyla MBRQ1, MBRQ2, MBRQ3 ve MBRQ4 adlı veri yapılarına yerleştirerek işletimini tamamlar. Ana yordamın uygun biçimde işleyebilmesi için söz konusu listeler zaman bilgisi ile birlikte saklanmaktadır.

Kavram düzeyi kod içerisinde SAL öneki ile belirtilen hizmetler uygulama programcısını pek çok alt-düzey ayrıntıdan yalıtacak biçimde tasarlanmıştır. Şu anda İB'ler arası iletişim için socket-tabanlı bir yapı kullanılmaktadır. Eğer gerekirse bu yapı daha sonra var olan uygulamalarda değişiklik gerektirmeksizin bir başka iletişim yapısıyla değiştirilebilir. *SAL_CommStructure* içinde tutulan socket tanımları uygulama programcısının erişimine açık olmadığından uygulama programı içerisinde doğrudan kullanılamaz. *GetNextFrame* komutu kendi işletim dizisine sahip olup ana işletim dizisi sorgulayana kadar resim karelerini biriktirmeye devam eder ve böylece uygulama yazılımcısını ara-bellek yönetimi yapmak yükünden kurtarır. *SVC_GatherResults* komutu ise işletim dizinlenmiş ayrı bir uygulamadır ve her işletim dizisi bir socket bağlantısını denetler. Bu nedenle komut ilk kez işletildiğinde socket bağlantılarının kurulmasından kaynaklan bir gecikme yaşanacaktır. Komut işletimini tamamladıktan sonra yaratılan işletim dizinleri sonlandırılmaz ve çalışmayı sürdürür. Bunların silinmesi ancak *SAL_CommStructure* yapısında yer alan her kaynağı geri alan *SAL_destroy* komutunun işletilmesi ile gerçekleşir. Bu soyutlama düzeyinde yer alan işlevlerin çoklu-çekirdekli işlemcilerin getirdiği olanakları kullanılabilmesi üzerinde araştırmamız sürmektedir.

Aktarmalı Hedef Takibi

Yukarıda açıklanan altyapı, kamera, görev ve İB gibi öğelerin dinamik olarak kullanıma alınıp kullanımdan çıkarılması gereken daha karmaşık uygulamaların gerçekleştirilmesi için de kullanılabilir. Örneğin verilen kavram düzeyi kodu, kolaylıkla izleme görevinin kameradan kameraya aktarıldığı bir insan takibi uygulamasına dönüştürülebilir. İzlenecek insanın, görüş alanı içinde bulunduğu kameranın görüş alanı dışına çıkacak biçimde hareket ettiğini varsayalım. Bu kişinin hangi kameraların görüş alanına gireceği ağ ilingesi kullanılarak kestirilebilir. Bu durumda ana yordam hemen bu kameralarla ilgilenecek bir İB grubu oluşturur ve izlenen kişinin nitelik vektörünü ya da matematik modelini ileterek İB'ler üzerinde gerekli görevleri başlatır. Hedef, bir kameranın görüş alanından diğer kameraların görüş alanına doğru hareket ettikçe görev parametreleri güncellenir ve İB grubuna gerekli ekleme ve çıkarmalar gerçekleştirilir. Burada her yeni görev tanımında *CameraID* parametresinin değişeceğine dikkat edilmelidir. Bu işlemler için

htd.TerminateTask, *hTD.ModifyTaskParam*, ve *htd.RemoveProcessingUnit* komutları kullanılmalıdır.

Görürsen Bildir

Akıllı güvenlik kamerası uygulamalarının tasarımında değişik programlama yaklaşımlarının kullanılması gerekebilir. Önceki örneklerde GAK tarafından sunulan hizmetler tek bir görevin iş yükünü birkaç İB birimine paylaşmak için kullanılmıştı. Bir başka deyişle daha güçlü ve pahalı bir bilgisayar yerine sıradan kişisel bilgisayarları çok sayıda kullanarak işlem-yoğun bir görevin yerine getirilmesi sağlanmıştı. Bu nedenle görevler arası iletişim kanalları görevler başlatılmadan önce açılmıştı. Bu yaklaşımın olay bildirim türündeki uygulamalarda etkin kaynak kullanımını sağlamayacağı açık olduğundan, önerilen sistem mimarisine olay-tabanlı bir mekanizma eklenmiştir. Buna örnek olarak, polisin sayısal resmini sisteme girdiği bir kişi için anakent genelinde arama başlatması gibi nitelik vektörü verilen bir nesne ya da kişinin aranmasına yönelik bir uygulama ele alınacaktır. İlgili kavram düzeyi kodu Şekil 4’de yer almaktadır.

```
FaceDetectMaster {
    hPUG = new (SAL_ReportBackGroup);
    hPUG.SetMasterProcessingUnit(PU5);
    hPUG.AddProcessingUnit(PU1);
    hPUG.AddProcessingUnit(PU2);
    hPUG.AddProcessingUnit(PU3);
    hPUG.AddProcessingUnit(PU4);
    // task definition commands remain the same
    Status = SAL_ActivateRBTasks(hComm);

    while (not done) {
        SVC_WaitForReport(hComm);
        ProcessResults(hComm);
    }
    SAL_destroy(hComm);
}
```

Şekil 4: Görürsen Bildir Uygulamasının Kavram Düzeyi Kodu

Arama kıstası ilgili tüm İB'lere dağıtılacak ve böylece kamera görüntülerinin eş zamanlı olarak taranması sağlanacaktır. Ancak bu durumda ana yordamın tüm İB'lerle işlemin hemen başında iletişim kurması anlamlı değildir. Bunun yerine Microsoft Windows programlamada kullanılan *CallBack Procedure* yapısına benzer bir yapı görev yapmaktadır.

Burada İB grubunun adı, *SAL_ReportBackGroup*, işlevini açıklar niteliktedir. İşlemi başlatan komut da öncekilerden farklı olup ana yordamla İB'ler arasında herhangi bir iletişim kanalı açmaz. Ancak tüm İB'ler bulgularını nereye ileteceklerini, bir başka deyişle grubun liderinin adresini bilmektedir.

SONUÇ

Geniş ölçekli güvenlik kamerası ağları, karmaşıklık düzeyi çok yüksek olan dağıtımlı sistemlerdir ve aynı oranda karmaşık bir sistem mimarisine gerek duyarlar. Kameranın, görüntünün işlendiği bilgisayara doğrudan bağlandığı geleneksel laboratuvar düzenekleri sözü edilen mimarinin yapıtaşları olarak kullanılacak niteli-

klere sahip değildir. Zaman içinde dağıtımlı sistem mimarilerinde geçerliklerini kanıtlamış yöntemler de işlenecek ve aktarılacak verinin boyutlarının çok büyük olması nedeniyle geniş ölçekli güvenlik kamerası ağlarına aynen uygulanamaz. Dolayısıyla, söz konusu sistemlerin, kendine özgü mekanizmalar ve teknikler geliştirilmesini gerektiren ayrı bir tür olarak ele alınması gerektiği görüşündeyiz. Bu bildiri de güvenlik uygulamaları geliştirme konusuna özellikle vurgu yapan bir mimari önerilmiştir. Önerilen mimari, görüntü alma, depolama ve görevler arası iletişim süreçlerinin alt düzeyli ayrıntılarından uygulama programcısını yalıtılmak üzere tasarlanmıştır. Uygulamaların daha hızlı ve verimli geliştirilebilmesini kolaylaştıran bu yaklaşım aynı zamanda alt-düzyer işlevlerin yazılan uygulamaları etkilemeden değiştirilebilmesine de olanak sağlamaktadır.

ÖZGEÇMİŞ(LER)



Harun ARTUNER 1982, yılında Ankara Üniversitesi Fizik Mühendisliği Bölümünden mezun olan Yrd. Doç. Dr. Harun ARTUNER, 1990-1994 yıllarında Hacettepe Üniversitesi Bilgisayar Mühendisliğinde Doktora çalışmasını tamamlamıştır. Aynı bölümde Öğretim Üyesi olarak çalışmaktadır. Aynı zamanda Hacettepe Üniversitesi Adli Bilişim Araştırma ve Uygulama Merkezi Kuruculuğunda bulunmuş ve yönetim kurulu üyesidir. Sayısal Sinyal İşleme, Konuşma Tanıma, Gömülü Sistemler, Yapay Anlayış ve Adli Bilişim konularında çalışmalarını sürdürmektedir.



Cesur BARANSEL. 1985 yılında Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümünden mezun olan Doç. Dr. Cesur Baransel, 1985–1990 yılları arasında Bilişim Ltd.'de programcı, sistem çözümleyici ve proje yönetmeni olarak çeşitli görevlerde bulunmuştur. 1990–1993 yılları arasında, Kanada'nın Alberta Üniversitesinde, Bilgisayar Ağları konusunda doktora çalışmasını tamamlayan Doç. Dr. Cesur Baransel, daha sonra Netaş'da araştırma mühendisi, Likom Yazılım'da projelerden sorumlu Genel Müdür Yardımcısı ve Saltus Yazılım'da Genel Müdür olarak görev yapmıştır. Hacettepe Üniversitesi Adli Bilişim Araştırma ve Uygulama Merkezinde danışman ve öğretim üyesi olarak katkı vermiştir.



Turhan MENTEŞ. 1955 Yalova doğumlu olan Menteş, 1977 Hacettepe Üniversitesi İstatistik Bölümünü bitirmiştir. 1984 yılında Doktor ve 1996 yılında da İstatistiksel Bilgi Sistemleri Ana Bilim dalında Doçent olmuştur. 2011 Yılında Profesör olan Menteş halen Hacettepe Üniversitesinde öğretim üyeliği ve Genel Sekreterlik görevine devam etmektedir. Bu görevini yürütürken eş zamanlı olarak 1993-1996 yılları arasında Türkiye Odalar ve Borsalar Bir-

liđi, Başkan Danışmanlığı ve TOBB Enformasyon ve Bilgisayar Ađı Kurulması Projesi Proje Yöneticiliđi, Tarım Orman ve Köy İşleri Bakanlığı, TUYAP Projesi, Bilgisayar Ađı Fizibilite Çalışmasında Yazılım Danışmanlığı, 1990-1991 yıllarında Yüksek Öğrenim Kurumunda Özel İhtisas Komisyonu üyeliđi yapmıştır. 1996 yılından beri MERNİS gibi birçok bilişim projesinde de proje yöneticiliđi yapmıştır. İstatistik Mezunları Derneđi'nde 1997-2000 dönemi yönetim kurulu başkanlığı, Türkiye Bilişim Vakfı kurucu üyeliđi, 1996-2000 yıllarında da İktisadi Kalkınma Vakfı denetçiliđi yapmıştır. 1993 yılından beri Türkiye Okçuluk Federasyonu yönetim kurulu üyeliđi ve as başkanlığını yürütmektedir. Türkiye Bilişim Derneđi'nde, 2005 yılından beri Yönetim Kurulu Başkanlığı görevini sürdürmektedir.

KAYNAKLAR

1. Aghajan, Hamid – Cavallaro, Andrea (2009), *Multi-Camera Networks, Principles and Applications*, Academic Press, Burlington, MA 01803, USA.
2. Bashir, Faisal – Porikli, Fatih (2007), “Collaborative Tracking of Objects in EPTZ Camera”, *SPIE Conference Visual Communications and Image Processing Vol. 6508*, January.
3. Cao, Xuesong – Wang, Zhaoping – Hu, Ruimin – Chen, Jun (2008), “An Overlay-Based Service Architecture for Distributed Video Surveillance System”, *International Journal of Software Engineering and Its Applications*, 2(2): 129–140.
4. Desurmont, Xavier – Delaigle, Jean-François – Bastide, Arnaud – Macq, Benoit (2004), “A Generic Flexible and Robust Approach for Intelligent Real-time Video-Surveillance Systems”, *Proceedings of SPIE-IS&T 16th Symposium on Electronic Imaging Science and Technology*, 134–141.
5. Detmold, Henry – Dick, Anthony – Falkner, Katrina – Munro, David S. – Van den Hengel, Anton – Morrison, Ron (2006), “Middleware for Surveillance Applications”, *ACM MidSens'06*, Australia, 31–36.
6. Enficiaud, Raffi – Lienard, Bruno – Allezard, Nicolas – Sebbe, Raphael – Beucher, Serge – Desurmont, Xavier – Sayd, Patrick – Delaigle JF (2006), “CLOVIS: A Generic Framework for General Purpose Visual Surveillance Applications”, *IEEE Workshop on Visual Surveillance*, 177–184.
7. Fonseca, A. – Mayron, L. – Socek, D. – Marques, O. (2008), “Design and Implementation of An Optical Flow-Based Autonomous Video Surveillance System”, *International Conference on Internet and Multimedia Systems and Applications (EuroIMSA2008)*, March 17–19, Austria.
8. Hampapur, Arun – Brown, Lisa – Connell, Jonathan – Pankanti, Sharat – Senior, Andrew – Tian, Yingli (2003), “Smart Surveillance: Applications, Technologies and Implications”, *Joint Conf. of the Fourth International Conference on Information, Communications and Signal Processing, and the Fourth Pacific Rim Conference on Multimedia*. 1133–1138.
9. McKee, P. (2008), “Why the French Spot Terrorists Better than Americans,” *Law Enforcement Technology*, January 2008, 92–95.
10. Migdal, Joshua – Izo, Tomas – Stauffer, Chris (2005), “Moving Object Segmentation using Super Resolution Background Models”, *The 6th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*.
11. Miguel, J.C.S – Bescos, Jesus – Martinez, Jose M. – Garcia, Alvaro (2008), “DiVA: A Distributed Video Analysis Framework Applied to Video-surveillance Systems”, *Proc. of Ninth Int. Workshop on Image Analysis for Multimedia*, 207–210.
12. Valera, M. – Velastin, S.A. (2005), “Intelligent Distributed Surveillance Systems: A Review”, *IEE Proc. Vision, Image and Signal Processing*, 192–204.
13. Wijnhoven, R.G.J – Jaspers, E.G.T. – de With, P.H.N. (2006), “Flexible Surveillance System Architecture for Prototyping Video Content Analysis Algorithms”, *Conference on Real-Time Imaging IX, Proceedings of the SPIE*.
14. Zhou, Xuhui – Collins, Robert T. – Kanade, Takeo – Metes, Peter (2003), “A Master-Slave System to Acquire Biometric Imagery of Humans at Distance”, *First ACM SIGMM international workshop on Video surveillance*, 113–120.
15. www.avigilon.com/products/samplefootage, 09.08.2011.

