

# Etkileşimli Parçacık Süzgeci Yöntemi ile Kapatmaya Dayanıklı Yüz ve El Takibi

## A Particle Filter Based Algorithm for Robust Tracking of Hands and Face Under Occlusion

Oya Aran, Lale Akarun

Bilgisayar Mühendisliği Bölümü  
Boğaziçi Üniversitesi, İstanbul, Türkiye  
{aranoya, akarun}@boun.edu.tr

### Özetçe

Bu çalışmada işaret dili haber videolarında el ve yüz takibi için parçacık süzgeci tabanlı bir algoritma geliştirdik. İşaret dilinde eller birbirini ya da yüzü kapatabilir. Bu tür durumlarda da gürbüz izleme yapabilecek bir izleme algoritmasına ihtiyaç vardır. Bu amaçla, eller ve yüz için ayrı parçacık süzgeçleri kullandık ve her süzgecin diğer süzgeçlerin parçacıklarının ağırlıklarını etkilediği bir algoritma önerdik. Bu yöntem ile hem bir nesneye ait olan parçacıkların diğer nesnelere sığmasını önledik hem de temas ya da kapatma durumunda izlemenin başarı ile devam etmesini sağladık.

### Abstract

This paper presents a particle filter based algorithm for tracking face and hands of a signer. During signing, the hands and the face occlude each other frequently and a proper multiple object tracking algorithm is needed for accurate results. We use separate particle filters for the two hands and the face, where each filter effects the particle weights of others. This methodology ensures that the particles of each object will not jump to other objects and the tracking continues without problem during occlusion.

### 1. Giriş

Doğal hızında gerçekleştirilen işaret dili videolarında, elleri ve yüzü herhangi bir belirteç kullanmadan takip edebilmek henüz çözülememiş bir problemdir. El ve yüz takibi için çeşitli algoritmalar önerilmiş olsa da bu algoritmaların büyük çoğunluğu doğal ortama uygulanması zor olan çeşitli ön kabuller içermektedir. Ellere çeşitli belirteçler konulması ya da ellerin ve yüzün birbirini örtmeyeceği kabulü bunlardan bazılarıdır. İşaret dilinde ise, dilin doğası gereği, eller birbirleriyle ya da yüzle sıklıkla temas halindedir. İşaret dili işlemede, ellerin ve yüzün yüksek doğrulukla takip edilmesi, işaret dili tanıma algoritmalarında yüksek performans elde edebilmek için ön koşuldur. Bu nedenle temas ve kapatmadan etkilenmeyecek bir izleme algoritmasına ihtiyaç vardır.

Nesne izleme için kullanılan algoritmalarından en bilineni Kalman süzgecidir. Kullanımının basit olması ve gerçek zamanlı çalışabilmesinden dolayı bir çok uygulamada başarıyla

kullanılmıştır. Kalman süzgeci izlenen nesnenin doğrusal hareket ettiğini ve sistemde Gauss gürültüsü bulunduğunu kabul eder. Doğrusal olmayan sistemler için Kalman süzgeci temelli, genişletilmiş Kalman süzgeci gibi algoritmalar önerilmiştir. Kalman süzgeci tabanlı algoritmalar, bir sonraki video karesi için nesnenin yeri hakkında tek bir kestirim yaparlar.

Parçacık süzgeçleri ise olasılıksal izleme algoritmalarıdır [1]. Nesnenin yeri hakkında birden çok kestirimde bulunurlar. Bu özellikleri sayesinde birden fazla nesneyi takip edebilme kapasiteleri vardır. Parçacık süzgeçleri özellikle doğrusal olmayan ve Gauss gürültüsünün geçerli olmadığı sistemlerde iyi performans göstermiştir. Koşullu Yoğunluk Yayılımı (CONDENSATION)[2] algoritması parçacık süzgecinin bir uygulaması olup, imge serisinde nesne izleme problemi için önerilmiştir. Parçacık süzgeci imge üzerine çeşitli parçacıklar atarak nesne hakkında tahminlerde bulunur. Bu yöntemle parçacıklar izlenen nesne ya da nesnelerin bulunduğu yer ya da yerlerde yoğunlaşırlar. Parçacık süzgeçleri birden fazla nesneyi takip edebilecek kapasitede olsalar da nesnelere birbirleriyle temas halinde olduğunda ya da birbirlerini kapattıklarında aynı başarıyı gösteremezler. Bu problemi çözmek için, değişken sayıda birden fazla nesneyi takip eden parçacık süzgeci temelli çeşitli yöntemler önerilmiştir [3, 4, 5].

Bu çalışmada çok sayıda nesnenin takibi sırasında temas ve kapatma durumlarında da gürbüz izleme yapabilmek için bir yöntem önerdik. Etkileşimli parçacık süzgeci, farklı nesnelere takip eden parçacık süzgeçlerinin birbirlerinin parçacıklarının ağırlıklarını etkilemesi sayesinde izlemeyi gerçekleştirir. Elleri ve yüzü basit bir elips modeli ile modelledik. Yöntemin başarımını, işitme engelliler haber bülteni görüntülerinden kaydedilmiş, toplam yedi dakikalık bir videoda test ettik ve yüz için % 100, eller için ise ortalama % 92 başarı elde ettik.

### 2. Parçacık Süzgeci

Parçacık süzgeci, Bayesçi yaklaşımı, ağırlıklandırılmış çok sayıda parçacık üzerinde uygulayarak sonsal dağılıma yakınsamaya çalışır. Bu parçacıklar izlenen nesnenin dinamiklerine göre hareket ettirilir ve imge üzerindeki olabirliklerine göre ağırlıklandırılır. İmge üzerine dağılan parçacıklardan olabirliği yüksek olanlar izlenen nesneyi en yakın olduğu

düşünülen parçacıklardır ve sistemde daha fazla kalırlar.

Parçacık süzgeçleri, diğer Bayesçi süzgeçler gibi, sonsal olasılık dağılımını bulmaya çalışırlar:

$$p(\mathbf{x}_t | \mathbf{x}_0, \mathbf{Z}_{1:t}) \quad (1)$$

$\mathbf{x}_t$  sistemin  $t$  anındaki durumunu,  $\mathbf{Z}_{1:t}$  de başlangıçtan  $t$  anına kadarki gözlemlerimizi ifade eder.  $\mathbf{x}_0$  izlemeyi başlatabilmek için gerekli olan ilk durum değeridir. Bu değer bir önsel olasılık dağılımına göre belirlenebilir. Parçacık süzgeci, bu sonsal dağılıma yakınsamak için,  $N$  adet ağırlıklandırılmış parçacık kullanır.

$$\{(\mathbf{x}_t^n, \pi_t^n) : n = 1, \dots, N\} \quad (2)$$

$\mathbf{x}_t^n$   $t$  anındaki  $n$  parçacığının durum vektörünü,  $\pi_t^n$  ise ağırlığını gösterir.

Parçacık süzgeci izleme problemini iki ana fonksiyon yardımıyla çözmeye çalışır: durum fonksiyonu ve ölçüm fonksiyonu. Durum fonksiyonu ( $f$ ),  $t - 1$  anındaki bir parçacığın,  $t$  anındaki durumunu belirler. Ölçüm fonksiyonu ( $h$ ) ise  $t$  anındaki bir parçacığın izlediğimiz nesne ile ne kadar örtüşüğünü hesaplar:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) \quad (3)$$

$$z_t = h(\mathbf{x}_t) \quad (4)$$

$f$  fonksiyonu izlenecek nesnenin dinamiğini modeller. Doğrusal olmayan harekete sahip nesnelere izleyebilmek için doğrusal olmayan bir  $f$  fonksiyonu seçilebilir.  $h$  fonksiyonu ise izlenecek nesnenin görünümünü modeller ve bir olabilirlik değeri döndürür. Parçacık süzgeçlerinin izleme başarısı kullanılan fonksiyonların izlenecek nesnenin dinamiğini ve olabilirliğini iyi modellemesine bağlıdır.

1. İklendirme:  $\{(\mathbf{x}_0^n, \pi_0^n)\}_{n=1}^N$
2. For  $t > 0$ 
  - (a) Yeniden örnekleme:  $\{(\mathbf{x}_{t-1}^n, \pi_{t-1}^n)\} \rightarrow \{(\mathbf{x}'_{t-1}, 1/N)\}$
  - (b) Öngörü:  $\mathbf{x}_t^n = f(\mathbf{x}'_{t-1})$   
 $\{(\mathbf{x}'_{t-1}, 1/N)\} \rightarrow \{(\mathbf{x}_t^n, 1/N)\}$
  - (c) Ağırlıklandırma:  $\pi_t^n \propto z_t^n = h(\mathbf{x}_t^n)$   
 $\{(\mathbf{x}_t^n, 1/N)\} \rightarrow \{(\mathbf{x}_t^n, \pi_t^n)\}, \sum_{n=1}^N \pi_t^n = 1$
  - (d) Kestirim:  $\hat{\mathbf{x}}_t = E\{(\mathbf{x}_t^n, \pi_t^n)\}$

Şekil 1: Koşullu yoğunluk yayılımı algoritması

Şekil 1'de parçacık süzgecinin basit bir uygulaması olan koşullu yoğunluk yayılımı [2] algoritması gösterilmiştir. Bu algoritmanın ilk adımı, önsel dağılım fonksiyonuna göre parçacıkların ilk değerlerini ve ağırlıklarını belirlemektir. Daha sonra, her  $t$  anı için sırasıyla, yeniden örnekleme, öngörme, ağırlıklandırma adımları gerçekleştirilerek  $t$  anındaki parçacıklar oluşturulur. Yeniden örnekleme adımında,  $t - 1$  anındaki parçacıkların ağırlıkları göz önünde tutularak, iadeli örnekleme ile eşit ağırlıklı yeni parçacık kümesi oluşturulur. Bu yöntem ile, bir önceki parçacık kümesinin yüksek ağırlıklı parçacıklarının yüksek olasılıkla bir sonraki adıma geçmesi

sağlanmış olur. Örneklenmiş parçacıkların  $t$  anındaki durumları,  $f$  durum fonksiyonu ile (Denklem 3),  $t$  anındaki ağırlıkları ise düzgelemlenmiş  $h$  ölçüm fonksiyonu ile (Denklem 4) belirlenir.

İzlediğimiz nesnenin  $t$  anındaki durumu,  $t$  anındaki parçacıkların ağırlıklı ortalaması ile kestirilebilir. Gerektiği durumlarda, bu ağırlıklı toplam sadece belli bir eşik değerini ( $\Phi_p$ ) aşan ağırlıktaki parçacıklar kullanarak da hesaplanabilir.

$$\varphi_t = \begin{cases} 1, & \text{eğer } \pi_t^n > \Phi_p \\ 0, & \text{eğer } \pi_t^n \leq \Phi_p \end{cases}$$

$$\hat{\mathbf{x}}_t = \frac{\sum_{n=1}^N \mathbf{x}_t^n \cdot \pi_t^n \cdot \varphi_t}{\sum_{n=1}^N \pi_t^n \cdot \varphi_t} \quad (5)$$

### 3. Etkileşimli Parçacık Süzgeci

Görüntüde benzer görünüm özelliklerine sahip birden fazla nesnenin izleneceği durumda, standart parçacık süzgeci algoritması iyi başarımlar gösteremez. Her ne kadar yakınsadığı sonsal dağılım çok kipli bir olasılık dağılımına imkan tanısa da, özellikle nesnelerin temas etmesi ya da birbirini kapatması durumunda dağılım tekrar tek kipli olur ve iki farklı nesne tek nesne olarak algılanır. Bu problemi çözmek için izlemek istediğimiz her nesne için ayrı parçacık süzgeçleri oluşturduk ve her nesnenin bir diğer nesnenin parçacıklarının ağırlıklarını etkileyerek temas ve kapatma durumlarında izlemeyi başarı ile gerçekleştirmesini sağladık.

Tek bir nesne için kullandığımız durum vektörü pozisyon, hız ve şekil parametrelerinden oluşmaktadır. Şekil parametrelerini, el ve yüzü bir elips ile modelleyerek seçtik. Bu model ile tek bir nesne için yedi boyutlu bir durum vektörü elde ettik.

$$\begin{aligned} \mathbf{x}_t^n &= [P_t^n, V_t^n, S_t^n]^T \\ P_t^n &= (x_t^n, y_t^n)^T \\ V_t^n &= (vx_t^n, vy_t^n)^T \\ S_t^n &= (a_t^n, b_t^n, \theta_t^n)^T \end{aligned} \quad (6)$$

$x_t^n, y_t^n$   $t$  anındaki  $n$  parçacığının merkezini,  $vx_t^n, vy_t^n$  hızını,  $a_t^n, b_t^n$  her iki elips ekseninin boyutlarını ve  $\theta_t^n$  açısını belirtmektedir.

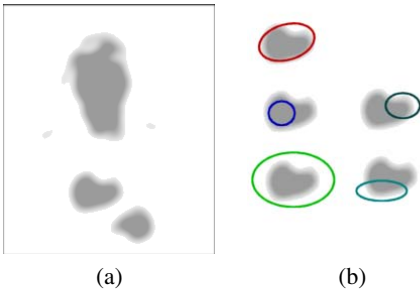
#### 3.1. Dinamik Model ve Olabilirlik modeli

Nesnenin pozisyon ve hızını sönümlü hız modeli ile, şekil parametrelerini ise rasgele yürüyüş ile modelledik:

$$\begin{aligned} V_t &= \lambda_v V_{t-1} + \sigma \mathcal{N}(0, 1) \\ P_t &= P_{t-1} + V_t \\ S_t &= S_{t-1} + \mathcal{N}(0, 1) \end{aligned} \quad (7)$$

Olabilirlik fonksiyonunu hesaplamak için öncelikle resimdeki ten renkli bölgeleri eğitilmiş bir Gauss karışım modeli kullanılarak belirledik [6]. Elde ettiğimiz gri seviye imgeyi önce Gauss yumuşatmasından geçirdik ve üst ve alt eşik değerleri kullanarak olabilirlik fonksiyonunu hesaplayacağımız imgeyi oluşturduk (Şekil 2a). Bu imge ten rengi olan pikseller için yüksek olasılık, ten rengi olmayan pikseller için ise sıfır olasılık değeri içermektedir.

Olabilirlik fonksiyonunu hem elipsin içinde kalan alandaki ten rengi piksellerin olasılığının yüksek olmasını, hem de



Şekil 2: (a) Olabilirlik hesabında kullanılan eşiklenmiş imge, (b) El bölgesi ve farklı parçacıklar. Olabilirlik fonksiyonu en üstteki parçacık için en yüksek değeri verir.

elipsin elin ya da yüzün dış sınırlarından geçmesini gerektiren şekilde seçtik. Elips sınırlarının olabilirlik fonksiyonu içine katılmadığı durumlarda küçük elipslerin ten rengi olmayan piksel içerme olasılığı çok daha az olduğu için parçacıklar zaman içinde küçülme eğiliminde olurlar. Dolayısıyla olabilirlik fonksiyonumuz elipsin dış halkasındaki piksellerin ortalama ten rengi olasılığı ( $A$ ) düşük olduğunda, içinde kalan piksellerin ortalama ten rengi olasılığı ( $B$ ) da yüksek olduğunda yüksek olabilirlik değeri döndürür.

$$z_t = \begin{cases} 0 & , eğer B < \Phi_p \\ 0.5 \cdot (1 - A) + 0.5 \cdot (B) & , eğer B \geq \Phi_p \end{cases} \quad (8)$$

Denklem 8'deki ilk satır, ten rengi piksel içermeyen ya da çok az içeren elipsler için düşük olabilirlik vermek için gereklidir. Aksi halde, bu tür elipslere hiç ten rengi piksel içermeseler bile 0.5 olabilirlik değeri verilir. Denklem en yüksek değerini, dış halkada hiç ten rengi piksel yoksa ( $A = 0$ ), ve iç kısımdaki tüm pikseller ten rengi pikselleri olduğu durumda ( $B = 1$ ) alır. Şekil 2b'de imge üzerinde değişik parçacıklar gösterilmiştir. En üstteki parçacık en yüksek olabilirlik değeri alırken, diğer parçacıklar daha düşük olabilirlik değeri alır.

### 3.2. Parçacık Etkileşimi

Bağımsız parçacık süzgeçleriyle izlediğimiz nesnelere temas ettiğinde ya da çok yaklaştıklarında bazı parçacıklar izlediğimiz nesneden uzaklaşıp diğer nesneye yaklaşabilirler. Bu parçacıkların izlemeye olan etkisini azaltmak için, Kang ve Kim [4] çalışmalarında rekabetçi parçacık süzgeci algoritmasını önerirler. Yürüyen insanları takip etmek için önerdikleri algoritmada nesnelere birine ait parçacıkların ağırlıkları, diğer nesneye ait parçacıklara olan uzaklıklarına göre yeniden belirlenir. Biz de el ve yüz izleme amacıyla benzer bir yaklaşımı uyguladık. Ellere ve yüze ait parçacık süzgeçleri parçacıklarının ağırlıklarını belirledikten sonra, parçacık etkileşimi ile parçacıkların ağırlıklarını güncelledik.

İlk aşamada, sol ve sağ el süzgeçlerinin parçacıklarının ağırlıklarını diğer elin kestirimine olan uzaklığını kullanarak güncelledik.

$$\begin{aligned} \pi_t^{n,r} &= \pi_t^{n,r} \cdot (1 - 1/\exp(\alpha \cdot \|\mathbf{x}_t^{n,r} - \hat{\mathbf{x}}_t^r\|)) \\ \pi_t^{n,l} &= \pi_t^{n,l} \cdot (1 - 1/\exp(\alpha \cdot \|\mathbf{x}_t^{n,l} - \hat{\mathbf{x}}_t^l\|)) \end{aligned} \quad (9)$$

Yukarıdaki formüllerde  $l, r$  indeksleri sol ve sağ ele ait



Şekil 5: (a) Elin yüzün önünde olduğu durumda izleme sonucu. (b) Yüz bölgesindeki piksellerin değişimi.

parçacıkları belirtmektedir. Şekil 4 ellerin temas halinde olduğu ardışık karelerde izleme sonucunu göstermektedir.

Aynı yöntemi yüze yakın parçacıklar için kullandığımızda, istenilen başarıyı vermediğini gördük. Bu nedenle yüz bölgesine belli bir uzaklıktan yakın olan parçacıkların ağırlığını düşürmek yerine sıfıra eşitledik.

$$\begin{aligned} \pi_t^{n,i} &= 0, eğer \|\mathbf{x}_t^{n,i} - \hat{\mathbf{x}}_t^f\| < d \\ d &\propto \hat{b}_t^f, i = r, l \end{aligned} \quad (10)$$

$d$  uzaklık eşik değerini belirtir ve yüze oturtulan elipsin küçük ekseninin uzunluğu ile orantılı olarak belirlenebilir.  $f$  indeksi yüze ait parçacıkları belirtmektedir.

### 3.3. El - Yüz Kapatması Durumunda İzleme

Parçacık etkileşimi, farklı nesnelere takip eden parçacık süzgeçlerinin parçacıklarının bir diğer nesneye yönelmemesini sağlar. Fakat, işaret dilinde el sıklıkla yüze temas eder ya da yüzün önünden geçer. Denklem 10 ile yaptığımız ağırlık güncellemesi ise buna izin vermemektedir. Bu tür durumlarda izlemeyi gerçekleştirebilmek için parçacıkların ağırlık güncellemesini yaparken elin yüzün önünde olup olmadığına dikkat etmek gerekir.

İşaret dilinde el hareketlerinin yanında kafa hareketleri de kullanılır. Haber videolarında ise, spiker görece olarak daha durağan bir şekilde işaretleme yapmaktadır. Dolayısıyla, videodaki iki ardışık kare arasında kafa hareketinin küçük olduğunu kabul edebiliriz. Bu ön kabulü kullanarak, iki ardışık karenin yüz bölgeleri arasındaki farkı hesapladığımızda, yüz bölgesindeki her pikselin değişimini elde ederiz. El yüz bölgesine girdiğinde, toplam değişim artacak, uzakta olduğunda ise değişim en alt düzeyde olacaktır. El yüz bölgesine girdikten sonra ise, iki ardışık kare arasındaki fark yerine el yüz bölgesine girmeden önceki son kareyi kullanmak gerekir. Aksi halde elde edilen piksel değişimi bilgisi sağlıklı olmayacaktır.

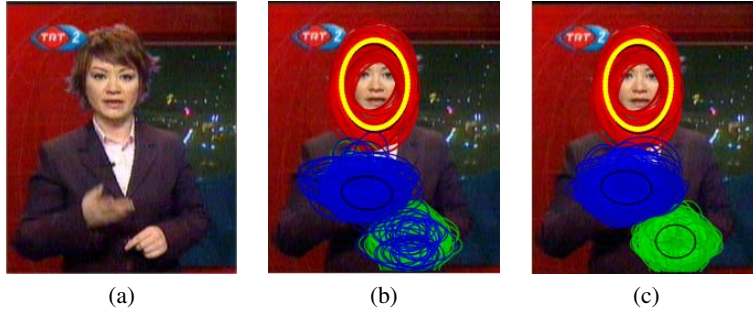
Bu bilgiyi de kullanarak, Denklem 10'u şu şekilde değiştirdik:

$$\pi_t^{n,i} = 0, eğer (\|\mathbf{x}_t^{n,i} - \hat{\mathbf{x}}_t^f\| < d \ \& \ M_{\mathbf{x}_t^{n,i}} < \Phi_m) \quad (11)$$

$M_{\mathbf{x}_t^{n,i}}$ ,  $\mathbf{x}_t^{n,i}$  parçacığının altında kalan bölgedeki ortalama hareketi,  $\Phi_m$  ise anlamlı hareket miktarını belirtir. Denklem 11 sadece el içermeyen yüz bölgelerindeki parçacıkların ağırlıklarını sıfırlar. Bu sayede elin yüzün önünden geçtiği durumlarda da izleme başarıyla gerçekleşir (Şekil 5).

## 4. Deneysel ve Sonuçlar

Yöntemin başarımını ölçmek için, toplam yedi dakikalık iki haber videosunda her iki elin ve yüzün yerlerini elle



Şekil 3: (a) Orjinal imge, (b) Etkileşim öncesi parçacık dağılımı, (c) Etkileşim sonrası parçacık dağılımı. Mavi parçacıklar sağ ele, yeşil parçacıklar ise sol ele aittir. Siyah elips ise parçacık süzgeçlerinin kestirimlerini göstermektedir.



Şekil 4: Ardışık karelerde ellerin teması ve birbirini kapatması sırasında izleme sonucu. Mavi ve yeşil elipsler algoritmanın kestirdiği sol ve sağ el bölgelerini göstermektedir.

Tablo 1: İşaretlenmiş veri ile karşılaştırma. Hata değerleri  $x$  ve  $y$  koordinatlarındaki piksel uzaklığı cinsinden verilmiştir.

	Hata ( $x, y$ )	# Yanlış			% Doğru
		Konum	Kabul	Red	
Yüz	3.0, 8.0	0	0	0	100
Sağ el	8.1, 9.5	714	41	23	778
Sol el	7.1, 8.4	835	63	33	931

işaretleyerek yer gerçekliği verisi oluşturduk. Bulduğumuz el ve yüz konumlarını işaretlenmiş veri ile karşılaştırarak ortalama hatayı bulduk. Sonuçlar Tablo 1’de gösterilmiştir.

Ortalama hata, işaretlenmiş konum ile bulunan konum arasındaki  $x, y$  uzaklıklarının nesnenin mevcut olduğu kare sayısına bölümü ile hesaplanmıştır. Bulunan ve işaretlenmiş noktalar arasındaki Euclid uzaklığı, bulunan elipsin eksenlerinden küçük olanından daha büyük ise o noktayı yanlış bulunmuş olarak kabul ettik. Yanlış pozisyon, bu tür noktaların sayısını, yanlış kabul, resimde izlenen nesne yokken, yanlışlıkla nesne bulunan karelerin sayısını, yanlış red ise resimde nesne varken nesne bulunamayan karelerin sayısını gösterir. Doğruluk oranı ise doğru bulunan karelerin sayısının toplam kare sayısına bölümü ile hesaplanmıştır. Deneyde kullanılan videoda 210x250 piksel çözünürlükte toplam 10332 kare bulunmaktadır. Bulunan elipslerin ortalama boyutları 30x30 civarındadır. Önerdiğimiz yöntem ile yüz tüm karelerde doğru bulunmuş, sağ ve sol el ise sırasıyla % 92.5 ve % 91.0 doğruluk oranı ile bulunmuştur.

## 5. Vargılar

Bu çalışmada önerdiğimiz yöntem, doğal hızında ve şeklinde yapılan işaret dili görüntülerinde gürbüz el ve yüz izlemesi yapmaya olanak tanımaktadır.

Nesne dinamiklerinin daha iyi kestirimi ve olabilirlik modelinin geliştirilmesiyle sistemin başarımının daha da artırılması mümkündür. Dinamik model, işaretlenmiş veri kullanılarak Saklı Markov Modeli ya da Kalman Filtresi ile belirlenebilir. İzlenen nesnelere temas ettikten sonra ayrılırken, bazı durumlarda parçacıkların yer değiştirdiği görülmektedir. Bu problem, daha yüksek çözünürlüklü imgeler ve daha ayrıntılı bir olabilirlik fonksiyonu kullanarak çözülebilir.

Bu çalışma Tübitak 107E021 nolu proje tarafından desteklenmektedir.

## 6. Kaynakça

- [1] Arnaud Doucet, Nando De Freitas, and Neil Gordon, Eds., *Sequential Monte Carlo methods in practice*, Springer-Verlag, 2001.
- [2] M. Isard and A. Blake, “Condensation – conditional density propagation for visual tracking,” *International Journal of Computer Vision*, vol. 26, no. 1, pp. 5–28, 1998.
- [3] M. Isard and J.P. MacCormick, “Bramble: A bayesian multiple-blob tracker,” in *International Conference on Computer Vision (ICCV’01)*, 2001, pp. II: 34–41.
- [4] H.G. Kang and D.J. Kim, “Real-time multiple people tracking using competitive condensation,” *Pattern Recognition*, vol. 38, no. 7, pp. 1045–1058, July 2005.
- [5] J. Czyz, B. Ristic, and B. Macq, “A particle filter for joint detection and tracking of color objects,” *Image and Vision Computing*, vol. 25, no. 8, pp. 1271–1281, August 2007.
- [6] Oya Aran, Ismail Ari, Pavel Campr, Erinc Dikici, Marek Hruz, Deniz Kahramaner, Siddika Parlak, Lale Akarun, and Murat Saraclar, “Speech and sliding text aided sign retrieval from hearing impaired sign news videos,” in *eNTERFACE’07 The Summer Workshop on Multimodal Interfaces, Istanbul, Turkey, 2007*.