|  |
| --- |
| **YILDIZ TEKNİK ÜNİVERSİTESİ – BİLGİSAYAR MÜHENDİSLİĞİ** |
| **Bilgisayarla Görme Final Ödevi** |
| **Mean Shift Algoritmasıyla Hareketli Nesne Takibi** |

|  |
| --- |
| Melike Nur Mermer - 15501010  13.06.2017 |

**Giriş:**

Bu çalışma Comaniciu tarafından önerilen Mean Shift algoritmasının [1] hareketli nesne takibi amacıyla kullanılmasını konu almaktadır. Algoritma üzerinde çalışılacak videonun frameler halinde ele alınmasıyla başlar, ilk framede takip edilecek nesne dikdörtgen içine alınarak belirlenir. Ardından bir sonraki frameden itibaren her framede mean shift algoritması uygulanarak ilk framede seçilen pencerenin histogramına önceden belirlenmiş yakınsama koşulunu sağlayıncaya kadar benzerliği en yüksek aday pencere belirlenir ve nesnenin yeni yeri olarak tespit edilir. Algoritma ball.avi ve test.avi videolarında denenmiş ve başarılı bir şekilde nesne takibini gerçekleştirdiği görülmüştür.

**Yöntem:**

Algoritmanın implementasyonunda [2]’deki kaynaktan istifade edilmiştir.

1. *Videonun Frameler halinde okunması ve İlk framede nesnenin seçimi*

Üzerinde çalışılacak video Matlab’in VideoReader() fonksiyonu ile okunmuş ve her framein bir eleman olarak tutulduğu struct dizisine kaydedilmiştir (Import\_mov.m). Bu dizinin ilk elemanında videonun ilk frame’i tutulmaktadır. Takip edilecek nesnenin seçimi için ilk frame gösterilir ve obje dikdörtgen içine alınır (Select\_patch.m fonk. getrect ile).

1. *Seçilen pencere boyutunda kernel oluşturulması*

Objenin çizilen dikdörtgenin ortasında olduğu varsayılıp ortadaki noktaların objeyi temsil gücünün daha yüksek olmasından yola çıkılarak çizilen dikdörtgen boyutlarında ağırlıklandırma penceresi oluşturulur. Pencere profili Uniform, Triangular, Epanechnikov veya Gaussian olabilir. Makalenin orijinalinde Epanechnikov pencere seçildiğinden bu çalışmada da bu profil kullanılmıştır. Ağırlık penceresinin oluşturulması için profilin yarıçapı da parametre olarak fonksiyona verilmektedir (Parzen\_window.m). Fonksiyonun çıkışında oluşturulan profil ile birlikte x ve y yönündeki gradyenleri de verilmektedir.

1. *Seçilen pencerenin histogramının çıkartılması*

Videoda RGB olan renkler indexler şeklinde ifade edilir. Seçilen pencere de videonun ilk frame’inde belirlenen renk haritasındaki indexler şeklinde ifade edilir. (Bu aşamadan sonra görüntüyü grayscale’de intensity değerleri ile ifade edilmiş gibi düşünebiliriz.) Bu işlemden sonra seçilen pencerenin renk histogramı çıkartılır (Density\_estim.m). Histogram çıkarma işleminde penceredeki her bir index histogramdaki bir bin’i oluşturmaktadır. Her bir bin’in değeri ise o index’e sahip olan noktaların parzen windowdaki (Epanechnikov) ağırlıklarının toplamıdır. Histogramın normalizasyonu için tüm binler parzen window ağırlıklarının toplamına bölünür.

1. *Hedefin video üzerinde gösterilmesi ve Takip için ön işlemler*

Çizilen dikdörtgen videonun ilk frameinde kırmızı çerçeve ile gösterilir (Draw\_target.m). Videonun ilk frame’i çerçeve çizilmiş hali olarak güncellenir. Her framede objenin yeri tespit edildikten sonra bu fonksiyon çağrılarak frame çerçeve çizilmiş hali ile güncellenmektedir. Bir sonraki frame de ilk framede belirlenen renk indexleri ile ifade edildikten sonra takip işlemine başlanabilir.

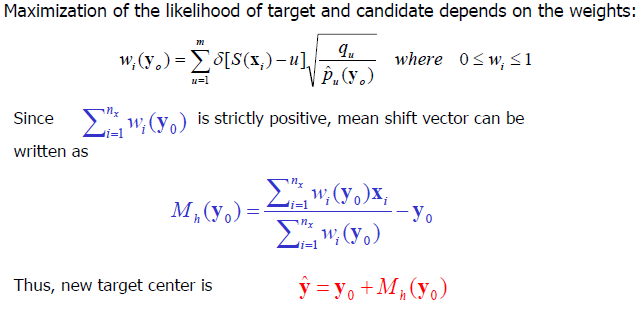
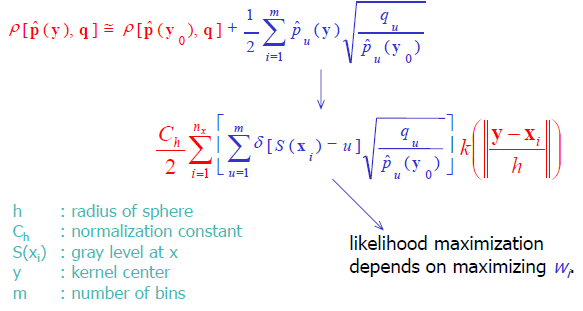
[1]Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis.", *IEEE Transactions on pattern analysis and machine intelligence* 24.5: 603-619*,* 2002.

[2]Mean-Shift Video Tracking by Sylvain Bernhardt, July 2008, <https://www.mathworks.com/matlabcentral/fileexchange/35520-mean-shift-video-tracking>

1. *Bir sonraki framede en benzer pencerenin bulunması için mean shift uygulanması*

Seçilen pencereye en benzer aday pencerenin bulunması için öncelikle ilk framede çizilen bölgenin sıradaki framedeki histogramı(p) çıkartılır. Bu benzerlik ölçümü yapılacak ilk adaydır. Aday pencerenin seçilen pencere ile benzerliği Bhattacharya Coefficient ile bulunur. Aday pencerenin benzerliği belirlenen eşik değerinden büyük olduğunda algoritma yakınsar.

Bu katsayının yüksek olması adayın seçilen pencereye benzer olması anlamına geldiğinden bir sonraki pencerenin seçimi için Taylor açılımındaki ağırlık ifadesinin maksimize edilmesi gerekir.



Her bir indexe ait histogram değerleri için bulunan ağırlık (w-histogram binlerinin birbirine bölünmesi(bin-by-bin)) o indexe sahip olan piksele karşılık gelen parzen window katsayısı ile çarpılır ve her bir piksel toplanarak benzerlik(f) bulunur ve normalize edilir (frame boyutu ile). Sıradaki aday pencereyi bulmak için mean shift algoritması uygulanır (pencere ortalamaya doğru kayar). Çerçevenin her bir pikselinin x ve y yönlerindeki kayma miktarlarının bulunması için koordinatlar w matrisi ve parzen window’un x ve y gradyenleri ile çarpılarak çarpımların toplamı ile normalize edilir. Mevcut başlangıç noktası kayma miktarları ile toplanarak yeni başlangıç noktası bulunur.

Yeni başlangıç noktasından çizilen pencereden çıkarılan histogram ile q histogramının benzerliği hesaplanır. w matrisi bulunur ve pencere yeni bulunan ortalama merkezine kayar. Belirlenen iterasyon sayısı boyunca pencere kaymaya devam eder (benzerlik eşiğini geçmediği sürece). Pencerenin en son durduğu yer nesnenin o frame’deki yeridir. Frame’de nesnenin çerçevesi çizilir ve bir sonraki framedeki ilk aday bu framedeki çerçeve olarak alınarak devam eder. Video’daki tüm framelerde obje bu şekilde takip edilir.

**Testler ve Sonuç:**

ball.avi üzerinde yapılan testler sonucunda farklı hızlara sahip topların farklı iterasyon sayılarında takip edilebildiği görülmüştür. Hızlı toplar için her framede daha fazla iterasyon boyunca takip etmek gerektiğinden daha yüksek iterasyon sayılarında daha iyi takip yapılabildiği görülmüştür. Ayrıca objeleri ifade etmede renk histogramları kullanıldığından objeler arka planı farklı bir alana geçtiklerinde takip zorlaşmaktadır. Yavaş top içinse yüksek iterasyon verilmesi çerçevenin hızla ilerleyip hedefi kaybetmesine yol açmaktadır.

test.avi üzerinde yapılan denemelerde bu video için Gaussian kernel’in Epanechnikov kernel’den daha iyi bir profil olduğu görülmüştür. Birbirine yakın duran benzer objelerden biri seçildiğinde (yan yana duran iki adam) diğeri ile karıştırılabildiği ve çerçevenin diğer objeyi takip edebildiği görülmüştür. Hedef olarak minibüs seçildiğinde aracın ön ucu görüntüden çıktığından hedef kaybedildi uyarısı görülmektedir. Aracın ön ucunun görüntüden çıktığı framedeki bir iterasyondaki aday histogramın çerçevesi videonun sınırlarının dışına çıktığından bu uyarı alınmaktadır.