

Artırılmış gerçeklik için BRIEF betimleyicileri ve yerelliğe duyarlı karma yöntemi ile nesne arama

Object detection with BRIEF descriptors and locality sensitive matching for augmented reality

Mustafa ÖZUYSAL^{1*}

¹Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, İzmir Yüksek Teknoloji Enstitüsü, İzmir, Türkiye.
mustafaouzysal@iyte.edu.tr

Geliş Tarihi/Received: 29.02.2016, Kabul Tarihi/Accepted: 22.02.2017

doi: 10.5505/pajes.2017.92604

* Yazışılan yazar/Corresponding author

Araştırma Makalesi/Research Article

Öz

Bu çalışmada mobil artırılmış gerçeklik için kullanılacak bir nesne arama yöntemi sunulmaktadır. Temel olarak yöntem anahtar nokta betimleyicilerinin eşleştirilmesine ve bu anahtar nokta eşlerinin geometrik kısıtlar ile süzülmesine dayanmaktadır. Eşlemenin hızlandırılması için gerekli iyileştirmeler detayları ile verilmektedir. Ayrıca, Yerelliğe Duyarlı Karma işleminin performansının bilgi erişim yaklaşımlarından faydalanılarak artırılacağı da gösterilmiştir.

Anahtar kelimeler: Bilgisayarlı görü, Nesne arama, Artırılmış gerçeklik, Yerelliğe duyarlı karma, İkili betimleyiciler

Abstract

In this paper, an object detection approach suitable for mobile augmented reality is presented. The baseline approach is based on matching keypoint descriptors and verifying these matches with geometric constraints. The performance optimizations necessary for speeding up matching are detailed. It is also demonstrated that it is possible to increase the performance of the Locality Sensitive Hashing by exploiting approaches from the information retrieval field.

Keywords: Computer vision, Object detection, Augmented reality, Locality sensitive matching, Binary descriptors

1 Giriş

Nesne arama yöntemleri, verilen bir görüntü üzerinde daha önceden bilinen bir nesnenin var olup olmadığının, var ise konumunun bulunmasını amaçlar. Artırılmış gerçeklik benzeri uygulamalarda sıkça kullanılırlar. Son yıllarda hem mobil işlemcilerin yaygınlaşması hem de işlem kapasitelerinin artması nedeniyle, mobil nesne arama yöntemleri de geliştirilmiştir [1].

Bu çalışmada gerçek zamanlı nesne arama için geliştirilen bir yöntem, detaylarıyla anlatılmaktadır. Nesne arama sonucunda bakış açısının detaylı hesaplanabilmesi ve nesnenin başka nesnelere kısmi örtülmesinden fazlaca etkilenmemesi için anahtar nokta tabanlı bir nesne arama yöntemi kullanılmıştır. Geliştirilen yöntem çeşitli aşamaları hızlandırılarak mobil cihazlarda çalışabilecek hıza eriştirilmiştir.

Anahtar nokta kullanılarak nesne arama yapılabilmesi için her bir anahtar noktanın etrafındaki örüntülerin öğrenilmesi gerekmektedir. Bu nedenle her nesne için bir eğitim görüntü setine ihtiyaç duyulur. Düzlemsel nesnelere için tek bir eğitim görüntüsü yeterli olduğundan, çalışmada da sadece düzlemsel nesnelere kullanılmıştır. Ancak, birden fazla görüntü üzerinden üç boyutlu bir model çıkarılarak [2], yöntemin üç boyutlu nesnelere de çalışması sağlanabilir. Nesne takip başarımının artırılması için ise sadece hedef nesnelere değil, ortamın da takibinin yapıldığı bir yöntem ile birleştirilebilir [3].

Anahtar nokta tabanlı nesne arama işleminin en hesaplamaya yoğun kısmı, görüntü üzerinde çıkarılan anahtar noktalar ile bilinen nesnelere eğitim görüntülerinden çıkarılan anahtar noktaların eşleştirilmesidir. Çalışmada bu eşleştirme işleminin hızlandırılması için Yerelliğe Duyarlı Karma [4] (YDK) yönteminden faydalanılmıştır. YDK yönteminin iyileştirilmesi için bilgi erişim (Information Retrieval) yaklaşımlarından esinlenilerek üç farklı hızlandırma yöntemi önerilmiştir.

Önerilen yöntemlerden ikisinin YDK yöntemine göre daha yüksek performans gösterdiği deneysel olarak gözlemlenmiştir.

Anahtar nokta eşleme yöntemi C/C++ dillerinde geliştirilmiştir. Bu sayede Android işletim sistemi için Java Native Interface (JNI) kullanılarak Java ile iOS işletim sistemi için ise doğrudan Objective C ile derlenen uygulamalara entegre edilebilmektedir. Nesnelere tanıtılması işlemi için ön işleme adımları (YDK için gerekli veri yapılarının oluşturulması) bir sunucu yardımıyla yapılarak mobil cihazda sadece bu veri yapılarının yüklenip kullanılması mümkündür.

2 Anahtar noktalar yardımı ile nesne bulma yöntemleri

Artırılmış gerçeklik uygulamalarında nesne konumunun görüntü içerisinde hassas bir biçimde belirlenmesi gerektiğinden nesne arama işlemi için anahtar nokta temelli yöntemler uygundur. Bu sayede pek çok anahtar noktadan elde edilen geometrik kısıtlar nesne/kamera göreceli konumunun tespitinde kullanılabilir. Anahtar nokta ile nesne arama yöntemleri pek çok alt adım içerir. Bunlar ile ilgili literatür özeti aşağıda verilmiştir.

2.1 Anahtar nokta bulma yöntemleri

Anahtar nokta bulma yöntemleri verilen bir görüntü üzerinde önemli noktaların tespitini amaçlar. Tespit edilen noktaların en önemli özelliği aynı sahnenin farklı kamera konum ve açısı, ışık seviyesi ve yönü ile çekilen görüntülerinde de büyük oranda tekrar bulunabilmeleridir. Bunun yanında noktaların çevresindeki örüntülerin ayırt edici olması ve noktaların görüntüde her yeri kaplayacak şekilde dağılması da istenen özelliklerdendir.

Anahtar nokta bulma yöntemlerinden en eski ve yaygın olarak kullanılanı Harris köşe bulma yöntemidir [5]. Bu yöntem ile

parlaklık eğiminin birbirine dik iki yönde de yüksek olduğu köşe benzeri noktalar bulunur. Bu sayede tespit edilen anahtar nokta etrafındaki örüntü, iki boyutlu hareketlerden çokça etkilenir ve dolayısıyla konum tespiti hassas bir şekilde yapılabilir.

Harris yöntemi iki boyutlu konumu net bir şekilde tespit ederken noktanın ölçeği hakkında bilgi vermez. Bu nedenle Lowe Ölçekten Bağımsız Öznitelik Dönüşümü (Scale Invariant Feature Transform, SIFT) yöntemini geliştirmiştir [6]. Bu yöntem ile anahtar nokta bulma işlemi Ölçek Uzayı Teorisine dayanır [7]. Verilen bir görüntünün önce Gauss piramidi hesaplanır. Ardından ardışık piramit seviyeleri birbirinden çıkarılarak bir Gaussların Farkı piramidi elde edilir. Piramit seviyelerinde hem iki boyutta hem de alt üst seviyelere göre yerel en yüksek değer elde edildiği noktalar anahtar nokta olarak seçilir. Daha sonra her bir anahtar nokta etrafındaki eğim doğrultularının dağılımlarındaki en büyük değere karşılık gelen yön anahtar noktanın doğrultusu olarak seçilir. Böylelikle hem konum, hem yön, hem de ölçek bilgisi saptanmış olur.

SIFT yöntemi ile birlikte ölçek ve doğrultu hesabının yapıldığı pek çok yeni anahtar nokta bulma yöntemi geliştirilmiştir [8], [9]. Ayrıca, bunların tekrarlanabilirlik açısından karşılaştırmaları da yapılmıştır [10]. Ancak, bu yöntemler görece hesap yoğun olduğundan mobil artırılmış gerçeklik için doğrudan kullanılmaları güçtür. Bu yüzden daha basit ve hızlı yöntemler geliştirilmiştir.

Bunlardan ilki SIFT yöntemine oldukça benzeyen ancak Ölçek Uzayı hesaplamasında Gauss filtreleri yerine Kutu (Box) filtreler kullanan SURF yöntemidir [11]. Kutu filtreler İntegral Görüntüler [12] ile çok hızlı hesaplanabildiğinden SURF anahtar noktaları SIFT'e göre daha hızlı hesaplanabilmektedir.

SIFT anahtar noktaları doğrudan mobil uygulamalarda tercih edilmese de parametreleri adapte edilerek cep telefonlarında artırılmış gerçeklik için kullanılmıştır [13].

FAST yöntemi piksel başına çok az sayıda ikili karşılaştırma kullanarak anahtar noktaları belirler [14]. Yöntem çok basit olduğundan ve anahtar nokta bulunmayan pikselleri bir iki karşılaştırma ile eleyebildiğinden oldukça hızlıdır. Özellikle mobil artırılmış gerçeklik uygulamalarında başarıyla kullanılabilir. Bu makalede detaylanan yöntem de FAST anahtar noktalarını kullanmaktadır.

2.2 İkilik betimleyiciler

Betimleyiciler, nesne arama amacıyla görüntü üzerinde tespit edilen anahtar noktaların nesnenin referans görüntülerinden çıkarılan anahtar noktalar ile eşlenmesi için kullanılan sayı dizileridir. Betimleyici vektörünün sayısal değeri, görüntülerin perspektif dönüşümlerinden çok fazla etkilenmemekte ve farklı anahtar noktaların etrafında hesaplanan betimleyici vektörleri de birbirlerinden farklı olmaktadır. Bu iki özellik sayesinde farklı görüntülerden hesaplanan iki betimleyici birbirine ne kadar yakınsa o oranda betimleyicilerin hesaplandığı anahtar noktaların aynı nesneye ait olduğu varsayılabilir.

SIFT yöntemi sadece anahtar nokta tespiti değil aynı zamanda betimleyici hesaplanmasında da kullanılır. SIFT betimleyicisi, bir anahtar nokta çevresindeki eğim doğrultularının dağılımlarının hesaplanmasına dayanır. Anahtar noktaları daha iyi ayırt edebilmek amacıyla tek bir dağılım yerine 4×4 boyutlarında on altı farklı bölgeden dağılımlar hesaplanarak uç uca eklenir. Elde edilen betimleyici vektörü L_2 normu 1

olacak şekilde ölçeklenir. Daha sonra sayısal değeri en büyük elemanı ve buna yakın elemanlarının değeri sabit bir değere çekilir ve betimleyici vektörü tekrar ölçeklenir. Bu son işlem, betimleyicinin ışık değişimlerinden daha az etkilenmesini sağlar.

SIFT betimleyicisi eşleşme başarısının önemli olduğu uygulamalarda yaygın olarak kullanılır. Ancak bu betimleyici, hesaplanması zaman aldığı için doğrudan gerçek zamanlı uygulamalarda kullanılmaya uygun değildir. Özellikle betimleyiciler arasındaki uzaklığın hesaplanması Öklid uzaklığı hesabını gerektirdiğinden görece işlem yoğunudur.

Gerçek zamanlı uygulamalarda kullanılmak üzere SIFT ve benzeri betimleyicilere alternatif olarak ikilik betimleyiciler geliştirilmiştir. Bunların en önemli avantajı iki betimleyici arasındaki uzaklığın sadece bir dışlamalı veya (XOR) ile ikil say (bit count) işlemleri ile hesaplanabilmesidir. Modern işlemcilerin Tekil İşlem Çoklu Veri (Single Instruction Multiple Data, SIMD) komut setleri sayesinde 256 ikil boyutundaki iki betimleyici arasındaki uzaklık sadece toplam dört makine komutu ile hesaplanabilir.

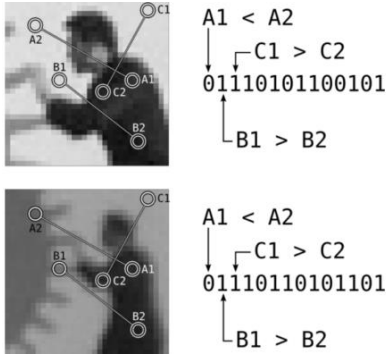
Görüntüler üzerinde doğrudan hesaplanan ilk ikilik betimleyici BRIEF betimleyicisidir [15]. BRIEF betimleyicisinin her bir ikil değerinin hesaplanması için anahtar nokta etrafında rastgele seçilen bir parlaklık değeri yine rastgele seçilen bir başka parlaklık değeri ile karşılaştırılır. İlk değer ikinciden büyükse betimleyicinin ikil değeri 1, aksi halde ise 0 olarak alınır. Bu şekilde 256 farklı karşılaştırma yapılarak betimleyicinin tüm ikil değerleri hesaplanır. Karşılaştırma için seçilen rastgele konumlar anahtar noktadan anahtar noktaya farklılık göstermez. Ayrıca karşılaştırmalardan önce görüntü bir Gauss filtresi ile işlendiğinden karşılaştırılan parlaklık değerleri tek bir noktanın parlaklığı değil o noktanın etrafındaki bir alanın ağırlıklı ortalama parlaklığıdır. Bu sayede hesaplanan ikil değerler görüntünün perspektif dönüşümlerinden fazlaca etkilenmez. BRIEF betimleyicisinin ve betimleyiciler arasındaki uzaklığın hesaplanması Şekil 1'de gösterilmiştir.

BRIEF betimleyicisinin dışında da pek çok ikilik betimleyici geliştirilmiştir [16]-[20]. Bu betimleyiciler ikilik değerleri farklı şekillerde hesaplarlar. Ancak BRIEF betimleyicisi gibi görece hızlı hesaplanabilirler ve betimleyiciler arasındaki uzaklık hesaplaması aynı verimliliktedir. Hesaplanma kolaylığı ve perspektif dönüşümlere dayanıklılığı nedeniyle bu çalışmada BRIEF betimleyicisi kullanılmıştır.

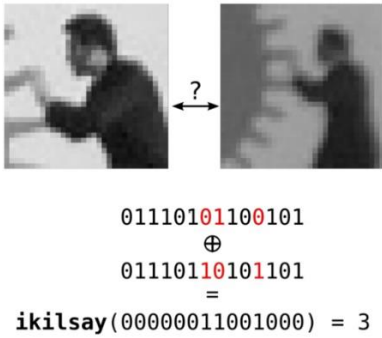
2.3 Yaklaşık en yakın komşu yöntemleri

Nesne arama sırasında, tespit edilmesi beklenen nesnenin üzerindeki anahtar nokta sayısı arttıkça, nesneye ait betimleyici sayısı da artar. Bu yüzden görüntülerde tespit edilen bir anahtar noktanın eşlenmesi için görüntüden çıkarılan betimleyicinin pek çok betimleyici ile uzaklığının hesaplanması gerekmektedir. Bu da nesne aramanın yavaşlamasına sebep olur.

Görüntüden çıkarılan bir betimleyiciye nesneye ait betimleyiciler arasında en yakın uzaklıktakinin seçimi için yaklaşık en yakın komşu yöntemleri kullanılabilir. Bu yöntemler sayesinde, nesne üzerindeki her bir betimleyiciye ayrı ayrı uzaklık hesaplanması gerekmez. Uzaklık sadece seçilen belirli bir betimleyici kümesi için hesaplanır ve en yakın komşunun bu küme içinde olduğu varsayılır.



(a): Betimleyicinin her bir ikil değeri farklı ve rastgele seçilen iki noktadaki parlaklık değerlerinin karşılaştırılması ile hesaplanır



(b): İki betimleyici arasındaki uzaklık sadece iki temel ikil fonksiyon kullanılarak hesaplanabilir.

Şekil 1: BRIEF betimleyicisi.

İkilik betimleyicilerin eşleştirilmesinde en çok kullanılan yaklaşık en yakın komşu yöntemi Yerelliğe Duyarlı Karma (YDK)'dır (Locality Sensitive Hashing, LSH) [4].

YDK yöntemi arama yapılacak tüm ikilik betimleyiciler için bir karma fonksiyonunun hesaplanması ve aynı karma değere sahip betimleyicilerin aynı grupta yer almasına dayanır. En yakın komşu hesaplanırken, aranan betimleyici için de karma değeri hesaplanır ve sadece bu karma değerine sahip betimleyiciler arasında arama yapılır. Tek bir karma fonksiyon ile yapılan gruplama ile gerçek en yakın komşunun bulunması olasılığı düşüktür. Bu olasılığı arttırmak için birden fazla karma fonksiyon kullanılır ve arama elde edilen her karma değerine karşılık gelen grupların tümünü kapsar.

Karma fonksiyon olarak ikilik betimleyicilerin düşük boyutlu uzaylara izdüşümleri kullanılabilir. D boyutlu ikilik betimleyiciler, $M \ll D$ seçilerek M boyutlu izdüşümlerine göre 2^M gruba ayrılırlar. Her betimleyicinin en yakın komşusu kendi grubu içinde arandığından çok daha az karşılaştırma yapılır. Bu işlem N kere rastgele seçilen boyutlar ile tekrarlanır ve N yeterince büyük seçilerek gerçek en yakın komşu ve aranan betimleyicinin büyük olasılıkla en az bir grupta birlikte yer alması sağlanır. M arttıkça gruplardaki ortalama betimleyici sayısı azalacağından arama hızı artar. Ancak, bu durumda gerçek en yakın komşuyu kaçırmamak için gruplama sayısını da arttırmak gerekir. Bu da eşleme işlemi yavaşlatır. Uygulamaya göre farklı M ve N değerleri seçilerek hız ve arama başarısı arasındaki uygun denge bulunabilir.

İkilik betimleyicilerin hızlı eşleştirilmesi için YDK dışında da yöntemler geliştirilmiştir [21],[22]. Ancak, bunlar görece daha

karmaşık kodlama gerektirir ve eşleme başarımında her zaman bir kazanç sağlayamazlar. Bu nedenle bu çalışmada YDK yöntemi kullanılmıştır. Anahtar nokta eşleme için geliştirilen hızlandırma yöntemlerinin genel bir özeti [23]'te verilmiştir.

Betimleyici eşlemesi için son zamanlarda yaygınlaşan iki yaklaşım Çarpım Nicemlemesi (Product Quantization) [24] ve çizge üzerinde yaklaşık en yakın komşu hesaplanmasıdır [25]. Ancak bu yöntemler henüz ikilik betimleyicilere uyarlanmadığından gerçek zamanlı mobil uygulamalarda kullanılmaları mümkün değildir.

3 BRIEF betimleyicisi ve yerelliğe duyarlı karma ile nesne arama

Anahtar noktalar ile nesne arama, bulunan nesnelerin görüntülerde detaylı geometrik verisi elde edilebildiği için artırılmış gerçeklik uygulamalarında sıkça kullanılır. Bu amaçla aşağıda detaylanan sistem geliştirilmiş, artırılmış gerçeklik uygulamalarında test edilmiş ve yerelliğe duyarlı karma yöntemi iyileştirilerek performansı artırılmıştır.

3.1 Nesne arama için anahtar nokta çıkarımı ve betimleyici hesaplanması

Arama yapılacak nesnelerin görünüşlerinin öğrenilmesi için daha önceden bir eğitim görüntü seti hazırlanması ve bu eğitim görüntülerinden nesneye ait anahtar nokta ve betimleyicilerin çıkarılması gerekmektedir. Bu çalışmada (ve genelde artırılmış gerçeklik uygulamalarında) sadece düzlemsel nesnelere çalışıldığından eğitim seti nesnenin yüzey normaline dik bakış açısından çekilmiş tek bir görüntüdür.

Arama yapılacak her nesne için bir eğitim görüntüsü ve bu görüntü üzerinde nesnenin sınırlarını belirleyen bir poligon sistemin temel girdisidir. Bu görüntü üzerinde yaklaşık bin adet FAST anahtar noktası çıkarılır. Nesne arama yapılacak görüntülerde nesne boyutu eğitim görüntüsüne göre küçük de olabileceğinden anahtar nokta bulma işlemi bir Gauss piramidi üzerinde yapılır. Her piramit seviyesinde ayrı ayrı bulunan FAST noktaların tamamı nesne anahtar nokta kümesini oluşturur.

Çıkarılan anahtar noktalardan poligon içerisinde kalanlar nesne üzerinde kabul edilir ve her biri için BRIEF betimleyicileri hesaplanarak nesne verisi olarak saklanır. BRIEF hesaplaması yapılmadan önce görüntülerin Gauss filtresinden geçirilmesi gerekmektedir. Ancak zaten anahtar nokta çıkarılırken bir piramit hesaplandığından bu piramit BRIEF hesaplamasında da kullanılabilir. Bu amaçla piramidin seviye sayısı anahtar nokta çıkarılacak piramit seviye sayısından daha fazla tutulur. Pratikte fazladan iki seviye hesaplanmıştır. Daha sonra L seviyesindeki bir anahtar noktanın BRIEF betimleyicisi çıkarılırken parlaklık değerleri $L + 2$ seviyesinden örneklenir. Bu sayede fazladan Gauss filtrelemesi yapılmadan betimleyici hesaplanabilir ve betimleyici hesaplaması hızlanır.

3.2 BRIEF betimleyicilerinin eşlemesi

Bu aşamada nesne arama için gerekli tüm veriler sistemde mevcuttur. Her gelen video karesinde de benzer şekilde anahtar nokta ve betimleyiciler bir Gauss piramidi yardımıyla hesaplanır. Daha sonra görüntü üzerindeki her bir betimleyiciye nesne üzerindeki en benzer betimleyicinin bulunması gerekmektedir. Betimleyici benzerliği

betimleyiciler arasındaki Hamming uzaklığı hesaplanarak elde edilir. Eğer $\mathbf{q} = [q_1, q_2, \dots, q_D]$ görüntü üzerindeki, $\mathbf{p} = [p_1, p_2, \dots, p_D]$ ise nesne üzerindeki bir betimleyici ise bunlar arasındaki Hamming uzaklığı,

$$H(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^D \delta(p_i \neq q_i) = \text{ikilsay}(\mathbf{p} \oplus \mathbf{q}) \quad (1)$$

şeklinde hesaplanır. Burada *ikilsay* verilen bir ikil vektörün değeri 1 olan elemanlarının sayısını vermektedir.

Görüntü üzerindeki her bir anahtar nokta, betimleyicisi nesne üzerindeki betimleyicilerden en yakınına ait anahtar nokta ile eşlenir. Bu sayede görüntü üzerindeki anahtar noktalar ile nesne üzerindeki noktalar arasında bir eşleme elde edilmiş olur. Ancak burada yapılan her eşleme doğru değildir. Bazı anahtar noktalar görüntünün arka planından çıkarılmıştır ve perspektif dönüşümler nedeniyle betimleyiciler arasındaki uzaklık tamamen güvenilir değildir.

3.3 Geometrik filtreleme

Yapılan eşlemedeki hataların ayıklanması için geometrik kistaslar kullanılabilir. Nesnelere düzlemsel olduğundan bunların farklı kamera açılarından görüntüleri de bir eşyazım (homography) matrisi ile ilişkilidir. Doğru eşlemeler bu eşyazım dönüşümü ile uyumlu iken yanlış eşlemeler uyumsuzdur. Eşyazım hesaplanması için Rastgele Örneklem Uyumluluğu (Random Sample Consensus, RANSAC) [26] yöntemleri kullanılabilir. Bu yöntemler ile eşlenen noktalardan dört çift seçilerek bir eşyazım dönüşümü hesaplanır ve tüm eşlemelerden bu eşyazım ile uyumlu olanlar sayılır. Eşleşen anahtar noktalar görüntü üzerindeki \mathbf{x}_g ve nesne üzerindeki \mathbf{x}_n ise ve test edilecek eşyazım dönüşümü H ise, uyumluluk kriteri

$$\|H\mathbf{x}_n - \mathbf{x}_g\|_2 < t_u \quad (2)$$

ile verilir. Burada, t_u uyumluluk için gerekli kabul edilebilir en yüksek uzaklık değeridir. Pratikte $t_u \sim 3$ piksel olarak alınmıştır.

Bu işlem defalarca tekrarlanarak uyumlu eşleme sayısı en yüksek olan eşyazım seçilir. Bu sayede hem görüntü ile nesne eğitim görüntüsü arasındaki eşyazım hem de doğru eşlenen noktalar hesaplanmış olur.

Geometrik filtrelemenin iyileştirilmesi için RANSAC yöntemine iki iyileştirme uygulanmıştır. Bunlardan birincisi görece basit RANSAC yöntemi yerine, Chum ve Matas tarafından geliştirilen PROSAC [27] yönteminin kullanılmasıdır. Eşleşen her nokta için betimleyici uzaklığı bilindiğinden uzaklığın küçük olduğu eşlemelerin doğru olma olasılığı da daha yüksektir. Bu yüzden rastgele dört eşleştirme seçilirken düşük betimleyici uzaklığına sahip betimleyicilere öncelik verilir. Pratikte eşlemeler betimleyici uzaklığına göre küçükten büyüğe doğru sıralanır ve rastgele örnekleme işlemine en tepedeki $K = 20$ eşlemeden başlanır. Her örneklemeden önce K bir artırılarak örnekleme yapılabilecek eşleme sayısı toplam eşleme sayısına ulaşana kadar yavaş yavaş artırılır. Bu sayede doğru eşyazım dönüşümüne daha kısa sürede ulaşılır. Eğer uyumlu eşleme sayısı $T = 30$ veya daha fazla olan bir eşyazım elde edilirse bu doğru eşyazım olarak seçilir ve örnekleme son verilir.

İkinci iyileştirme ise rastgele örneklemeden elde edilen eşyazım dönüşümünün hızlı olarak ön elenmesidir. Geometrik filtreleme sırasında hesaplanan her eşyazım ve tüm eşlemeler

için uyumluluk hesabı yapılması uzun süre alır. Bazı eşyazım dönüşümleri nesnenin ayna görüntüsüne ya da çok eğik kamera açılarına karşılık gelir. Pratikte bu tür dönüşümlerin hemen hemen hepsi yanlış eşleme barındıran örneklemelerden gelir. Bu yüzden uyumluluk analizi yapılmadan bu tip dönüşümler reddedilebilir. Bu da arama işlemini hızlandırır. Bu tip dönüşümlerin tespiti için elde edilen eşyazım dönüşümü bir karenin (hesaplama kolaylığı nedeniyle kenar uzunlukları iki birim olan ve sol alt köşesi $(-1, -1)$ koordinatlarında olan kare) köşelerine uygulanır. Önce elde edilen dörtgen köşelerinin sıralaması kontrol edilerek bir ayna dönüşümüne karşılık gelip gelmediği test edilir. Eğer bir ayna dönüşümü yoksa o zaman iç eksenleri arasındaki açı ölçülür ve açı değeri belirlenen sınırlar dışındaysa örneklem ve eşyazım dönüşümü uyumluluk sayısını hesaplamadan reddedilir. Bu adım PROSAC işlem süresini yaklaşık yarı yarıya kısaltmaktadır. Eğer görüntülerde bir ayna dönüşümü varsa bu hızlandırma adımı uygulanamaz.

3.4 Nesne aramanın sonuçlandırılması

Geometrik filtrelemeden sonra elde edilen eşyazım ve bu eşyazım ile uyumlu eşlemeler, nesne arama işleminin son basamağının girdisidir. Öncelikle uyumlu eşleme sayısı belirli bir sınırın altında ise nesne arama başarısız olarak sonuçlanır ve nesnenin görüntü içerisinde yer almadığı sonucuna varılır.

Uyumlu eşlemelerin sayısının yeterli olduğu durumda, eşlenen tüm anahtar noktaların koordinatları kullanılarak tek bir eşyazım hesaplanır. Bu hesaplama Doğrudan Doğrusal Dönüşüm (Direct Linear Transform) [28] yöntemi ile yapılır. Hesaplanan eşyazım dönüşümü nesnenin eğitim görüntüsü ile arama yapılan görüntü arasındaki ilişkiyi tanımladığından nesnenin konumu da elde edilmiş olur.

Anahtar noktalar ile nesne aramanın temel adımları Şekil 1 ve **Hata! Başvuru kaynağı bulunamadı.**'de gösterilmiştir.

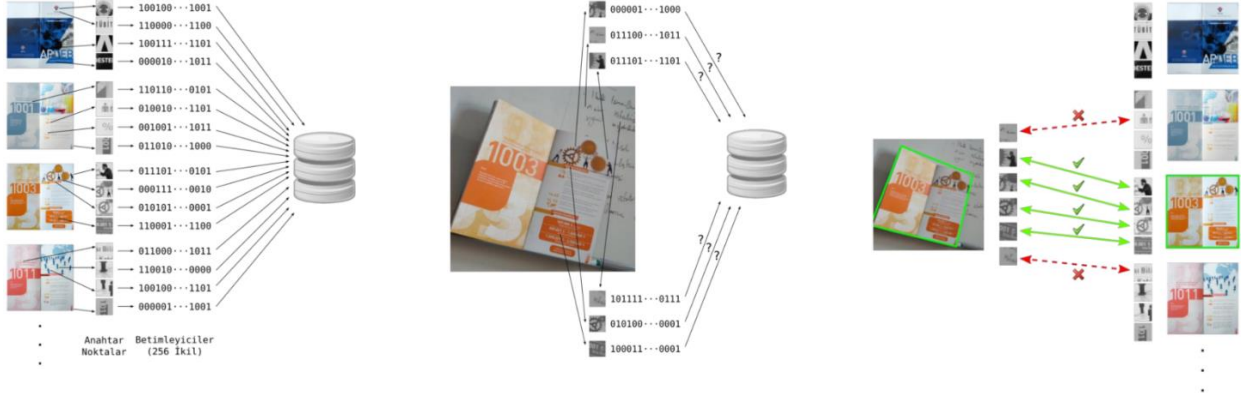
3.5 Yerelliğe duyarlı karma ile eşleme işleminin hızlandırılması

Nesne üzerindeki anahtar nokta, dolayısıyla betimleyici, sayısının görece az olduğu durumlarda Bölüm 3.2'de detaylanan eşleme işlemi tüm nesne betimleyicileri ile uzaklık hesaplayarak gerçekleştirilebilir. Ancak nesne betimleyicilerinin sayısı arttıkça yaklaşık en yakın komşu yöntemlerini kullanmak performans açısından daha uygundur. Bu nedenle eşleme işleminin hızlandırılması için de Yerelliğe Duyarlı Karma yöntemi kullanılmıştır.

Öncelikle nesne üzerinde yer alan anahtar noktaların betimleyicileri belirlenen sayıda karma fonksiyondan geçirilerek gruplanır ve hangi betimleyicinin hangi grupta olduğu verisi de saklanır. Eşleme sırasında verilen bir betimleyiciye en yakın betimleyici karma fonksiyon değerleri aynı olan her bir grup içinde aranır.

YDK yönteminin hızlandırma aşamasında önemli olan iki parametresi, kullanılan karma fonksiyon sayısı ve her karma fonksiyon için hesaplanan rastgele izdüşümün ikil sayısıdır. Önemli performans kistasları da tüm betimleyiciler içinde aramaya göre hızlanma oranı (λ) ve gerçek en yakın komşunun hangi oranda doğru olarak bulunabildiğini gösteren doğruluk değeridir (α). Bu performans değerleri şu şekilde hesaplanır:

$$\lambda = \frac{\text{YDK ile arama süresi}}{\text{Tüm betimleyicilerde arama süresi}} \quad (3)$$



(a): Öğrenilecek nesnelerin eğitim görüntüleri üzerinde anahtar noktalar tespit edilir, betimleyiciler hesaplanır ve bu verilir bir nesne veri tabanında saklanır.

(b): Arama yapılacak görüntü üzerinde tespit edilen anahtar noktaların betimleyicileri nesne veri tabanındaki betimleyicilerden en yakınları ile eşlenir.

(c): Eşlemeler geometrik olarak uyumluluk testine tabi tutularak doğru nesnenin hangisi olduğu ve konumu hesaplanır.

Şekil 2: Geliştirilen yöntemin adımları.

$$\alpha = \frac{\text{Gerçek en yakın komşunun bulunma sayısı}}{\text{En yakın komşu arama sayısı}} \quad (4)$$

Performans kriterleri ve YDK parametreleri arasındaki ilişki Şekil 2'de detaylandırılmıştır.

Karma fonksiyonda kullanılan ikil sayısı (M) arttıkça betimleyicilerin ait olabileceği grup sayısı da (2^M) artar. Böylelikle gruplar içerisindeki betimleyici sayısı azalır ve hızlanma artar. Ancak gerçek en yakın komşu ile aranan betimleyicinin aynı gruba düşmeme ihtimali arttığından doğruluk değeri azalır. Karma fonksiyon sayısı (N) arttıkça, arama yapılan betimleyici sayısı artar ve hızlanma azalır. Ancak, karma fonksiyonlardan herhangi biri için gerçek en yakın komşunun bulunması doğru arama sonucu için yeterli olduğundan, karma fonksiyon sayısının artması doğruluk değerini de artırır.

Pratikte hangi doğruluk ve hızlanma değerlerinin hedefleneceği nesne aramanın amacı ve kullanılan donanıma göre değişeceğinden nesne üzerindeki farklı betimleyici sayıları, M ve N parametreleri için ön testler yapılarak kullanılacak parametre değerleri seçilir. Doğruluk değerleri ölçülürken ikilik betimleyiciler için en yakın komşunun tek olmayabileceği de hesaba katılmalıdır.

4 Yerelliğe duyarlı karmanın hızlandırılması

Anahtar noktaların eşleştirilmesi için YDK yönteminin kullanılması ile birlikte arama hızı da artmaktadır. Ancak bazı karma fonksiyon değerleri için pek çok betimleyici bu değerlere sahip olduğunda arama süresi uzar. Çünkü bu betimleyiciler içerisinde arama sorgu betimleyici ile tek tek uzaklık hesaplanmasını gerektirmektedir.

YDK yöntemi ile bilgi erişim için kullanılan ters dizinler arasında benzerlikler bulunmaktadır [29]. Bu nedenle bu çalışmada bilgi erişim yaklaşımlarının hızlandırılmasında kullanılan üç farklı yöntem YDK için uyarlanmış ve betimleyici arama için kullanılması üzerine aşağıdaki yöntemler önerilmiştir. Deneysel test sonuçları ise Bölüm Hata! Başvuru kaynağı bulunamadı. de yer almaktadır.

4.1 Otomatik sorgu genişletmesi

Otomatik sorgu genişletmesi metin belgeleri aranırken arama yapılan terim kümesinin otomatik olarak eşanlı ya da benzer kelimeler kullanılarak genişletilmesidir. Anahtar noktalar için de birbiri ile küçük geometrik dönüşümler ile elde edilen betimleyicilerin benzer olacağı öngörülebilir. Sorgu için tek bir betimleyici kullanmak yerine betimleyici çıkarılan anahtar nokta için farklı dönme açılarında birden fazla betimleyici çıkarılıp her birinin ayrı ayrı sorgulanması denenmiştir. -20 ve +20 derece dönme açıları arasında 1-20 arası fazladan betimleyici hesaplanarak her bir karma fonksiyon tekrar tekrar kullanılmıştır. Bu durumda bir karma fonksiyon için arama süresi uzarken doğruluk değerinin artması ve daha az karma fonksiyon kullanılması ile hızlanma hedeflenmektedir.

4.2 Yalancı ilgililik geri beslemesi

Yalancı ilgililik geri beslemesi, arama sonucu bulunan metinlerin içeriğinin ikinci bir sorguda tekrar kullanılmasıdır. Bu sayede sorgunun ilgili terimler ile genişletilmesi mümkündür. Anahtar nokta sorgusu için bir karma fonksiyon kullanılarak elde edilen en yakın komşular diğer karma fonksiyonlarda sorgulama için kullanılabilir. Her tabloda 1 ile 15 arası en yakın komşu diğer tablolarda aranarak yine doğruluk değerinin artması ve daha az sayıda karma fonksiyon kullanılarak hızlanmanın artırılması hedeflenmiştir.

4.3 Gereksiz kelimelerin elenmesi

Gereksiz kelimelerin elenmesi metin aramalarında çok sayıda belgede geçmesi beklenen terimlerin arama listesinden çıkarılması işlemidir. Anahtar noktalar için de belirli bir sayıda betimleyiciyi de geçen karma fonksiyon değerlerini tamamen elenmesi ile hızlanma değerlerinin artırılması hedeflenmiştir. Bu durumda gerçek en yakın komşunun değeri elenen karma fonksiyon değerlerinden birine denk geldiğinde doğruluk değeri düşecektir. Doğruluk değerinde meydana gelecek azalmayı karşılamak için ise çok daha fazla tablonun kullanılması denenmiştir.

5 Deneyler

Geliştirilen nesne arama yöntemi, şablon (template) tabanlı bir takip yöntemi ile birlikte artırılmış gerçeklik uygulamasında kullanılmıştır. Daha sonra kullanılan YDK yönteminin iyileştirilmesi için Bölüm 4'te önerilen hızlandırma yöntemleri denenmiştir. Bu bölümde geliştirilen artırılmış gerçeklik uygulamasının niteliksel ve niceliksel bazı deney sonuçları yer almaktadır.

5.1 Artırılmış gerçeklik uygulaması

Anahtar nokta tabanlı nesne arama yöntemleri, yöntem sonucu olarak aranan nesnenin eğitim görüntüleri ile kamera görüntüleri arasındaki eşyazım dönüşümü hesaplandığından artırılmış gerçeklik uygulamaları geliştirmekte kullanılabilir. Bölüm 3'te detaylanan yöntem de bu amaçla kullanılmıştır.

Uygulama düzlemsel hedef nesnelerin üzerine iki boyutlu grafik ve animasyonların eklenmesini desteklemektedir. Hedef nesnelerin düzlem normaline dik bir açıdan elde edilen görüntüleri eğitim görüntüsü olarak alınmakta ve artırılmış gerçeklikte kullanılacak grafik ve animasyonlar bu eğitim görüntüsü üzerinde hazırlanmaktadır. Düzlemsel nesnelere için düzlem normaline dik tek bir görüntü diğer tüm açılardan elde edilen görüntülere bir eşyazım dönüşümü ile ilintili olduğundan bu görüntüden elde edilen anahtar noktalar diğer görüntülerde de büyük oranda tespit edilebilmektedir. Ancak düzlem normalinden 30-40° uzaklaşılınca anahtar nokta sayısı azalmaktadır. Eğer uygulama bağlamı gerektiriyorsa birden fazla görüntü eğitim verisine eklenebilir.

Nesne arama tamamlandığında bu işlem sonucu hesaplanan eşyazım dönüşümü kullanılarak grafik ve animasyonlar, eğitimi görüntüsü üzerindeki koordinatlarından, kamera görüntüleri üzerine taşınmaktadır. Geliştirilen nesne arama yöntemi, mobil cihazlarda da yaklaşık saniyede 10-15 kare işleyebilecek kapasitededir. Saniyede 35-45 kare işleyebilen bir nesne takip sistemi ile beraber, mobil artırılmış gerçeklik uygulamalarında kullanılabilir. Uygulamanın örnek çıktısı Şekil 3'te gösterilmiştir.

5.2 Yerelliğe duyarlı karmanın hızlandırılması

YDK yönteminin hızlandırılması için Bölüm 4'te önerilen yöntemlerin yanında, literatürde önerilen çoklu-sorgu [30] yöntemi de bulunmaktadır. Çoklu-sorgu yapılırken sadece aranan betimleyici ile değil bu betimleyiciye uzaklığı bir birim olan tüm betimleyiciler de aranır. Bu sayede tek bir karma fonksiyondan birden fazla defa yararlanır ve hafızadan tasarruf edilmiş olur. Bu yöntem önerilen otomatik sorgu genişletmesi yöntemine kavramsal olarak benzese de çoklu-sorgu için genişletme işlemi sadece geometrik uzaklığa bağlıdır. Temel YDK testleri yapılırken çoklu-sorgu için uzaklık değerleri sıfır (çoklu-sorgu kapalı) ile iki arasında değiştirilmiştir.

Temel YDK yöntemi BRIEF betimleyici veri tabanlarında farklı YDK parametreleri ile test edilmiş ve doğruluk ile hızlanma değerleri ölçülmüştür. Tanınacak görüntüler için Oxford veri setindeki "Wall" ve "Graffiti" test imgeleri kullanılmıştır (<http://www.robots.ox.ac.uk/~vgg/research/affine/>). Test veri tabanındaki toplam betimleyici sayısını arttırmak için bunlara Paris imge eşleştirme kümesindeki imgelerden çıkarılan BRIEF betimleyicileri eklenmiştir (<http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/>). Veri tabanında test imgeleri için yaklaşık 1000 BRIEF betimleyicisi bulunmakta ve tüm veri kümesi 100000

betimleyiciden oluşmaktadır. Aynı veri tabanında önerilen hızlandırma teknikleri ile de ölçüm yapılmıştır. Sonuçlar Şekil 4'te gösterilmiştir.



(a): Eğitim görüntüsü ve hedef sınırlarını belirleyen poligon.



(b): Farklı bakış açılarından tespit edilen görüntünün sanal grafikler ile zenginleştirilmesi.



(c): Farklı bakış açılarından tespit edilen görüntünün sanal grafikler ile zenginleştirilmesi.

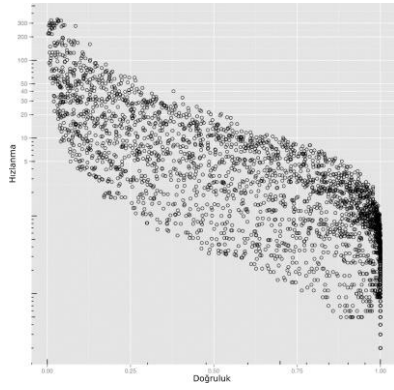


(d): Farklı bakış açılarından tespit edilen görüntünün sanal grafikler ile zenginleştirilmesi.

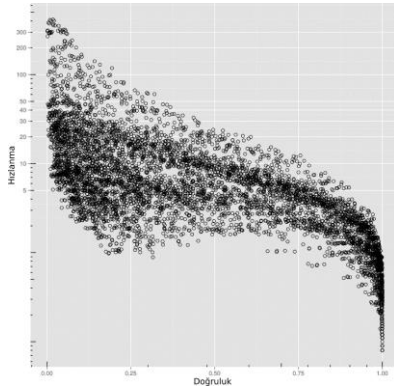


(e): Anahtar nokta tabanlı nesne arama yöntemleri, nesnenin kısmi olarak örtülmesinden de görece az etkilenmektedir.

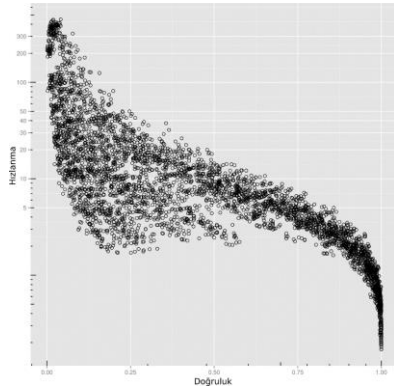
Şekil 3: Artırılmış gerçeklik uygulaması.



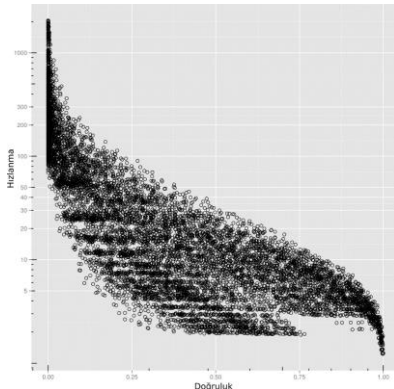
(a): Yerelliğe duyarlı karma.



(b): YDK ve otomatik sorgu genişletmesi.



(c): YDK ve yalancı ilgililik geri beslemesi.



(d): YDK ve gereksiz kelimelerin elenmesi.

Şekil 4: Temel yöntem YDK ile farklı hızlandırma yöntemlerinin pek çok karma fonksiyon sayısı ve izdüşümü büyüklüğündeki başarımının karşılaştırması.

Şekildeki her bir nokta kullanılan belirli bir M ve N parametre kombinasyonuna karşılık gelmektedir. Yine bazı noktalar çoklu-sorgu ile bazıları ise çoklu-sorgu olmadan yapılan testlere karşılık gelir. Farklı doğruluk oranları için elde edilen yaklaşık hızlanma değerleri karşılaştırmalı olarak Tablo 1'de görülebilir. Genel olarak bakıldığında önerilen hızlandırma yöntemlerinden en başarılısının gereksiz kelime elemesine dayandığı görülmektedir. Yalancı ilgililik geri beslemesine dayanan yöntem ise sadece YDK kullanımına göre daha başarısız olmuştur. Otomatik sorgu genişletmesine dayalı yöntem temel YDK yöntemine göre daha iyi sonuçlar elde etmiştir. Gereksiz kelime elemesine göre daha az hızlanma elde edilse bile araştırma ve geliştirmeye en açık yöntem de budur. Çünkü betimleyici benzerliği için dönme açısından farklı geometrik değişimler de önerilebilir.

6 Tartışma ve sonuç

Bu çalışmada anahtar noktaların görüntüler arasında hızlı bir şekilde eşleştirilmesine dayalı bir nesne arama sistemi detaylandırılmıştır. Nesne arama yönteminin en yakın komşu arama ve geometrik filtreleme adımları için iyileştirmeler önerilmiştir. Bu sayede sistem hem masaüstü hem de mobil cihazlarda artırılmış gerçeklik için başarıyla kullanılabilir.

Ayrıca ikilik betimleyiciler ile beraber sıkça kullanılan Yerelliğe Duyarlı Karma (YDK) yönteminin hızlandırılması için özgün yöntemler geliştirilmiştir. Özellikle otomatik sorgu genişletmesi ve gereksiz kelimelerin elenmesine dayalı yaklaşımların YDK yöntemine göre başarımının daha yüksek olduğu gözlenmiştir. Önerilen yöntemler YDK yönteminin hızlandırılması için sıkça kullanılan çoklu-sorgu yöntemine göre de daha başarılı olmuştur.

Mevcut sistem görüntülerde aynı anda sadece tek ve belirli bir nesneyi takip edebilmektedir. Bu da beş ila on arasında hedef nesnenin bulunduğu basit artırılmış gerçeklik uygulamalarında, ya da aranacak nesne sayısının coğrafi konum gibi bilgiler yardımıyla sınırlandırıldığı durumlarda kullanılması için yeterlidir. Ancak daha büyük nesne veri tabanlarında çalışabilmesi için görüntüdeki anahtar noktalardan olası hedef nesnelerin çıkarılması gerekmektedir.

Çalışmada imgelerdeki geometrik yani bakış açısına bağlı değişimlerden etkilenmeyen bir nesne tanıma sistemi geliştirilmiştir. Ancak, kamera hareketi ve ışık yetersizliği gibi nedenlerden ötürü oluşabilecek radyometrik bozulmalar sistemin çalışmasını etkileyebilir. Özellikle anahtar nokta bulma yöntemleri bu tip bozulmalara dayanıksız olduğundan eşleme adımında yapılacak müdahaleler yetersiz olmaktadır. Bu durumlarda anahtar nokta yerine doğrudan görüntü parlaklıkları üzerinde çalışan yöntemlerin [31] kullanılması daha uygundur. Yöntem üzerinde ileride bu tip iyileştirmelerin yapılmasına çalışılmaktadır.

Geliştirilen sistem sadece hedef nesnelerin takibi ile sınırlıdır. Artırılmış gerçekliğin tam anlamıyla uygulanması için daha kuvvetli bir yaklaşım ise ortamın bütünüyle haritalanıp takip edilmesi yaklaşımıdır [32],[33]. Böylelikle kamera hareketinden kaynaklı titremeler en aza indirilmiş olur ve konum bazı uygulamalar geliştirilebilir.

7 Teşekkür

Bu makalede yer alan çalışmalar TÜBİTAK tarafından 113E496 numaralı araştırma projesi kapsamında desteklenmiştir.

Tablo 1: Farklı doğruluk değerleri için YDK ile önerilen hızlandırma yöntemleriyle elde edilen yaklaşık en yüksek hızlanma değerleri. Önerilen yöntemlerden en başarılısı *gereksiz kelimelerin elenmesine* dayalı yaklaşımdır ve elde edilen hızlanma değerleri YDK yönteminin bir buçuk katı ve üstündedir. *Yalancı ilgililik beslemesine dayalı* yaklaşım için hızlanma sonuçları sadece YDK yönteminin kullanılmasına göre daha düşüktür. Otomatik sorgu genişletmesine dayalı yöntemin hızlanma başarısı YDK yönteminin üstünde olduğu için bu yöntem de pratikte YDK yöntemi ile çoklu-sorgu yerine kullanılabilir.

Doğruluk	Keskinlik			
	YDK	Otomatik Sorgu Genişletmesi	Yalancı İlgililik Geri Beslemesi	Gereksiz Kelimelerin Elenmesi
%50	20	25	16	30
%75	8	14	7	14
%87.5	5	6	4	8

8 Kaynaklar

- [1] Ondruska P, Kohli P, Izadi S. "Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones". *IEEE Transactions on Visualization and Computer Graphics*, 21(11), 1251-1258, 2015.
- [2] Özuysal M, Lepetit V, Fleuret F, Fua P. "Feature harvesting for tracking-by-detection". *European Conference on Computer Vision*, Graz, Austria, 7-13 May 2006.
- [3] Liu H, Zhang G, Bao H. "Robust keyframe-based monocular SLAM for augmented reality". *International Symposium on Mixed and Augmented Reality*, Merida, Mexico, 19-23 September 2016.
- [4] Andoni A, Indyk P. "Near optimal hashing algorithms for approximate nearest neighbor in high dimensions". *Communications of the ACM*, 51(1), 117-122, 2008.
- [5] Harris C, Stephens M. "A combined corner and edge detector". *Alvey Vision Conference*, Manchester, United Kingdom, 31 August - 2 September 1988.
- [6] Lowe D G. "Distinctive Image Features from scale-invariant keypoints". *International Journal of Computer Vision*, 20(2), 91-110, 2004.
- [7] Lindeberg T. "Scale-Space theory: A basic tool for analyzing structures at different scales". *Journal of Applied Statistics*, 21(1-2), 225-270, 1994.
- [8] Mikolajczyk K, Schmid C. "An affine invariant interest point detector". *European Conference on Computer Vision*, Copenhagen, Denmark, 28-31 May 2002.
- [9] Matas J, Chum O, Martin U, Pajdla T. "Robust wide baseline stereo from maximally stable extremal regions". *British Machine Vision Conference*, Cardiff, United Kingdom, 2-5 September 2002.
- [10] Mikolajczyk K, Schmid C. "A Performance evaluation of local descriptors". *IEEE Transactions on Pattern Analysis and Machine Learning*, 27(10), 1615-1630, 2004.
- [11] Bay H, Ess A, Tuytelaars T, Gool L V. "SURF: Speeded up robust features". *Computer Vision and Image Understanding*, 10(3), 346-359, 2008.
- [12] Viola P, Jones J. "Robust real-time face detection". *International Journal of Computer Vision*, 57(2), 137-154, 2004.
- [13] Wagner D, Reitmayr G, Mulloni A, Drummond T, Schmalstieg D. "Pose tracking from natural features on mobile phones". *International Symposium on Mixed and Augmented Reality*, Cambridge, United Kingdom, 15-18 September 2008.
- [14] Rosten E, Porter R, Drummond T. "Faster and better: A machine learning approach to corner detection". *IEEE Transactions on Pattern Analysis and Machine Learning*, 32(1), 105-119, 2010.
- [15] Calonder M, Lepetit V, Özuysal M, Trzcinski T, Strecha C, Fua P. "BRIEF: Computing a binary local descriptor very fast". *IEEE Transactions on Pattern Analysis and Machine Learning*, 34(7), 1281-1298, 2012.
- [16] Leutenegger S, Chli M, Siegwart R Y. "BRISK: Binary robust invariant scalable keypoints". *International Conference on Computer Vision*, 2011.
- [17] Rublee E, Rabaud V, Konolige K, Bradski G. "ORB: An efficient alternative to SIFT or SURF". *International Conference on Computer Vision*, Barcelona, Spain, 6-13 November 2011.
- [18] Alahi A, Ortiz R, Vanderghenst P. "FREAK: Fast Retina Keypoints". *Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, 16-21 June 2012.
- [19] Trzcinski T, Christoudias M, Lepetit V. "Learning Image Descriptors with Boosting". *IEEE Transactions on Pattern Analysis and Machine Learning*, 37(3), 597-610, 2015.
- [20] Levi G, Hassner T. "LATCH: Learned Arrangements of Three Patch Codes". <http://www.openu.ac.il/home/hassner/projects/LATCH> (29.02.2016).
- [21] Muja M, Lowe DG. "Fast matching of binary features". *Computer and Robot Vision Conference*, Toronto, Canada, 27-30 May 2012.
- [22] Trzcinski T, Lepetit V, Fua P. "Thick boundaries in binary space and their influence on nearest neighbor search". *Pattern Recognition Letters*, 33(16), 2173-2180, 2012.
- [23] Muja M, Lowe DG. "Scalable nearest neighbor algorithms for high dimensional data". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11), 2227-2240, 2014.
- [24] Kalantidis Y, Avrithis Y. "Locally optimized product quantization for approximate nearest neighbor search". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, United States of America, 24-27 June 2014.

- [25] Harwood B, Drummond T. "FANNING: Fast approximate nearest neighbour graphs". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, United States of America, 26 June - 1 July 2016.
- [26] Fischler M, Bolles R. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Communications of the ACM*, 24(6), 381-395, 1981.
- [27] Chum O, Matas J. "Matching with PROSAC-Progressive sample consensus". *Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 20-26 June 2005.
- [28] Hartley R, Zisserman A. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge, UK, Cambridge University Press, 2000.
- [29] Manning C, Raghavan P, Schütze M. *Introduction to Information Retrieval*. 1st ed. New York, United States of America, Cambridge University Press, 2008.
- [30] Lv Q, Josephson W, Wang Z, Charikar M, Li K. "Multi-Probe LSH: Efficient indexing for high-dimensional similarity search". *International Conference on Very Large Databases*, Vienna, Austria, 23-27 September 2007.
- [31] Forster C, Pizzoli M, Scaramuzza D. "SVO: Fast semi-direct monocular visual odometry". *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 31 May - 7 June 2014.
- [32] Engel J, Schöps T, Cremers D. "LSD-SLAM: Large-scale direct monocular SLAM". *European Conference on Computer Vision*, Zurich, Switzerland, 6-12 September 2014.
- [33] Arth C, Pirchheim C, Ventura J, Schmalstieg D, Lepetit V. "Instant outdoor localization and slam initialization from 2.5d maps". *IEEE Transactions on Visualization and Computer Graphics*, 21(11), 1309-1318, 2015.