İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

YEREL ÖZNİTELİKLER KULLANILARAK GÖRÜNTÜ İNDEKSLEME VE EŞLEME

YÜKSEK LİSANS TEZİ

Onur ÇALIKUŞ

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Bilge GÜNSEL

NİSAN 2016

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

YEREL ÖZNİTELİKLER KULLANILARAK GÖRÜNTÜ İNDEKSLEME VE EŞLEME

YÜKSEK LİSANS TEZİ

Onur ÇALIKUŞ
(504111353)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Bilge GÜNSEL

NİSAN 2016

İTÜ, Fen Bilimleri Enstitüsü'nün 504111353 numaralı Yüksek Lisans / Doktora Öğrencisi **Onur ÇALIKUŞ**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"YEREL ÖZNİTELİKLER KULLANILARAK GÖRÜNTÜ İNDEKSLEME VE EŞLEME"** başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Prof. Dr. Bilge GÜNSEL** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Prof. Dr. Muhittin GÖKMEN** MEF Üniversitesi	.....
	Yrd. Doç.Dr. Yusuf YASLAN İstanbul Teknik Üniversitesi	.....

Teslim Tarihi : **30 Mart 2016**
Savunma Tarihi : **28 Nisan 2016**

Eşime ve kızıma,

ÖNSÖZ

Yüksek Lisans süresi boyunca beni akademik konularda çalışmaya sevk eden, bana her konuda yardımcı olan değerli tez danışmanım Prof. Dr. Bilge Günsel'e sonsuz teşekkürlerimi sunarım. Kurumsal hayatta çalışmalarıma desteğini esirgemeyen Dr. Cenk Sezgin'e teşekkür ederim. Çalışmanın başında çalışmada benden önce emeği geçen Özgün Çırakman ve Sezer Kuluk'a ve veritabanı konusunda çalışmamıza ışık tutan Hayri Turgut Uyar'a teşekkür ederim. Benden desteğini esirgemeyen, en az benim kadar bu sürecin stresini çeken eşime, bu tezin oluşmasında büyük katkı sağlayan kardeşime, tüm öğrenim hayatım boyunca bana yol gösterici olan ve sabırla devam etmemi sağlayan anneme, her zaman arkamda olduğunu hissettiğim tecrübeleri ile bana ışık tutan babama, bu süreçte ihmal ettiğim sevdiklerime, dostlarıma ve akrabalarıma varlıklarından dolayı teşekkür ederim.
Bu tezin oluşabilmesi için gerekli tüm çalışmalar İTÜ Elektronik ve Haberleşme Mühendisliği Bölümü bünyesinde yer alan "Çoğulortam Sinyal İşleme ve Örüntü Tanıma Laboratuvarı"nda gerçekleştirilmiştir.

Nisan 2016

Onur Çalıkuş
Elektronik ve Haberleşme
Mühendisi

İÇİNDEKİLER

Sayfa

ÖNSÖZ ..... **vii**
İÇİNDEKİLER ..... **ix**
KISALTMALAR ..... **xi**
ÇİZELGE LİSTESİ ..... **xiii**
ŞEKİL LİSTESİ ..... **xv**
ÖZET ..... **xvii**
SUMMARY ..... **xix**
1. GİRİŞ **1**
2. YEREL ÖZNİTELİKLERLE GÖRÜNTÜ İÇERİK MODELLEME **11**
2.1 Görüntü Özniteliklerinin Elde Edilmesi 12
2.1.1 Ölçek Uzayı Uçdeğer Kestirimi 12
2.1.2 Anahtar Nokta Konumlandırma 15
2.1.3 Anahtar Noktalara Bağımsız Yönelimler Atanması 16
2.1.4 Anahtar Nokta Tanımlayıcıları (Öznitelik Vektörü) Oluşturulması 17
2.2 Görüntülerin Öznitelikleri Kullanılarak Eşleme Yapılması 18
3. İÇERİK TABANLI MOBIL GORUNTU ARAMA EŞLEME AĞACI **21**
3.1 Veri tabanı Modelleri 21
3.2 Sözcük Ağacı Oluşturma ve Indeksleme 23
3.2.1 Hiyerarşik k-ortalamalı kümeleme ile sözcük ağacı oluşturulması 24
3.2.2 Düğüm Indeks Bilgileri 27
4. MOBİL GÖRÜNTÜ ARAMA VE EŞLEME ALGORİTMALARI **31**
4.1 Görsel Benzerlik Skorlama 32
4.2 Ters İndeks (Inverted Index) 32
4.3 Bağlamsal Ağırlıklandırma (Contextual Weighting) 34
4.4 Önerilen Arama ve Eşleme Yöntemi 36
5. BAŞARIM TESTLERİ VE SONUÇLAR **39**
5.1.1 Veri tabanında tutulacak özniteliklerin indekslenmesi 39
5.1.2 İndekslenen özniteliklerden ağaç hiyerarşisinin belirlenmesi 40
5.1.3 Veri tabanında tutulacak ağaç yapısının oluşturulması 41
5.1.4 Veri tabanı oluşturma 41
5.2 Veri tabanı Üzerinde Görüntü Dağılımı 42
5.3 Görüntü Eşleme Başarımı Uygulama Alanı 46
5.4 Önerilen Arama ve Eşleme Yöntemi Sonuçlarının Detaylı Yorumu 50
6. SONUÇLAR **53**
KAYNAKLAR **57**
ÖZGEÇMİŞ **61**
..... **61**

KISALTMALAR

BoF **:** Bag of Features
CD **:** Compact Disc
DoG **:** Difference of Gaussian
DVD **:** Digital Versatile Disc
FAST **:** Fast Affine Template Matching
JDK **:** Java Development Kit
OODB **:** Object Oriented Database
RANSAC : Random Sample Consensus
RDBMS **:** Relational Database Management System
SDK **:** Software Development Kit
SIFT **:** Scale-Invariant Feature Transform
SURF **:** Speeded Up Robust Features
TF-IDF **:** Term Frequency – Inverse Document Frequency
XML **:** Extensible Markup Language

ÇİZELGE LİSTESİ

Sayfa

Çizelge 5.1 : Veri tabanlarındaki referans görüntülerin dağılımına istinaden oluşan katman, öznitelik düğüm ve yaprak düğüm sayıları **42**
Çizelge 5.2 : Nokia 5800 ile çekilen sorgu görüntüsünün seviye bazlı yaprak düğüm dağılımı **43**
Çizelge 5.3 : Nokia 5800 ile çekilen referans görüntüsünün seviye bazlı yaprak düğüm dağılımı **44**
Çizelge 5.4 : Nokia 5800 ile çekilen sorgu görüntüsü ile en iyi eşlene beş görüntü . **45**
Çizelge 5.5 : Sözcük ağacı eşleme başarım sonuçları. **46**
Çizelge 5.6 : Ters indeks eşleme başarım sonuçları. **47**
Çizelge 5.7 : Bağlamsal ağırlıklandırma eşleme başarım sonuçları. **47**
Çizelge 5.8 : Önerilen Yaprak Düğüm Ağırlıklandırma Eşleme Başarım Sonuçları. **48**

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Bütünsel (holistic) öznitelik ile yapılan eşleme (Zhang, 2015)..... **3**
Şekil 1.2 : Yerel (local) öznitelikler ile yapılan eşleme (Zhang, 2015)..... **3**
Şekil 1.3 : Mobile Visual Search Data Set örnekleri (Chandrasekhar, 2011). **5**
Şekil 1.4 : Üst satır bütünsel, ikici satır yerel ve birlikte kullanımı (Zhang, 2015). ... **8**
Şekil 2.1 : CD kapağı SIFT öznitelik eşleme örneği. **12**
Şekil 2.2 : Soldaki Gauss ile konvüle etme, sağda Gauss çıkarılmış DoG görüntü. . **14**
Şekil 2.3 : Uçdeğer belirlemede kullanılan komşuluk ilişkisi **14**
Şekil 2.4 : Öznitelik vektörü oluşturma..... **17**
Şekil 3.1 : k=3, solda üstten görünüm, sağda ağaç görünümü (Nister, 2011). **25**
Şekil 3.2 : Sözcük ağacı oluşturma görsel anlatımı (Nister, 2006). **26**
Şekil 3.3 : Özniteliklerin ağaç düğümlerine yerleştirilmesi **27**
Şekil 3.4 : Ağaç düğümlerine yerleşmiş sorgu görüntü öznitelikleri **28**
Şekil 4.1 : Hiyerarşik k-means kümeleme (Girod, 2011),..... **32**
Şekil 4.2 : Ters indeks dizisi (Girod, 2011),..... **33**
Şekil 4.3 : Öznitelik bağlamsal ağırlıklandırma örneği (Wang, 2011)..... **35**
Şekil 4.4 : Öznitelik bağlamsal ağırlıklandırmanın etkisi (Wang, 2011). **36**
Şekil 5.1 : "Stanford Mobile Visual Dataset" veri seti (Chandrasekhar, 2011) **39**
Şekil 5.2 : Cluster ve KeyPoint sınıflarının ağaç yapısında görsel gösterimi **42**
Şekil 5.3 : Nokia 5800 telefonla çekilmiş bir kitap sorgu ve referans görüntüsü..... **44**
Şekil 5.4 : Nokia 5800 telefonla çekilmiş en iyi eşlenen beş görüntü. **45**
Şekil 5.5 : Sorgu ve 001.jpg referans görüntüsü yaprak düğüm dağılımı **45**
Şekil 5.6 : Bahsedilen dört yöntem için başarım grafiği **49**
Şekil 5.7 : Bahsedilen dört yöntem için başarım grafiğinin sütun hali..... **50**
Şekil 5.8 : Farklı cihazlar ile farklı nesnelerin eşleme başarım oranı..... **51**

YEREL ÖZNİTELİKLER KULLANILARAK GÖRÜNTÜ İNDEKSLEME VE EŞLEME

ÖZET

Bu tez çalışmasında, yüksek başarım oranına sahip ve hesaplama karmaşıklığı düşük bir mobil görüntü arama, eşleme ve getirme sistemi geliştirilmesi amaçlanmaktadır. Bu amaç doğrultusunda çeşitli görüntü getirme ve eşleme metotları değerlendirilmiş ve farklı metodolojilerin farklı koşullar altındaki başarımını belirlemek için bu metotların performansları karşılaştırılmış ve yeni bir arama eşleme yöntemi önerilmiştir.

Görüntü öznitelikleri kullanılarak büyük ölçekli görüntü getirme, özellikle görüntü arama olmak üzere bir çok yeni gelişen uygulamalar sayesinde ilgi çekici araştırma alanlarından biridir. Görüntü getirme algoritmaları iki kategoriden oluşmaktadır. Bu kategoriler, ayrıt edici yerel öznitelikler ve kompak hash kodları ile indekslenen global özniteliklerdir. Bu tez çalışmasında, mobil arama amacıyla bir görsel görüntü getirme algoritması önerilmektedir. Bu tip uygulamlarda sorgu görüntüsü, mobil cihazın kamerası ile çekilmiş görüntüde yer alan yerel bir paterndir. Yerel özniteliklerin bölgesel görüntü paternlerini temsil edebilmesi sayesinde benzer görüntülerin veya bölgelerin ayırt edilmesinde güçlü bir yöntem olan SIFT öznitelikler ile sözcük ağacı indekleme yöntemi tarif edilmiştir.

Literatürde, sözcük ağacı indeksleme ve getirme alanında bir çok yöntem mevcuttur. Sözcük ağacı ile görüntü eşleme alanındaki öncü çalışma Nister'in çalışmasında bahsi geçen sözcük ağacı ile ölçeklenebilir eşleme isimli yöntem olarak kabul edilebilir (Nister, 2006). Bu çalışmadan esinlenerek Philbin (Philbin, 2007) sözcük ağacı yaklaşımını daha da geliştirerek, getirilen görüntülerin SIFT özniteliklerinin sorgu görüntüsüne benzerliğinin RANSAC ile kontrol edilmesini önermiştir. Chen'in çalışmasında (Chan, 2010) getirme hızının arttırılması ve sözcük ağacı indekslemesinde bellek ihtiyacının azaltılması amacıyla Nister'in (Nister, 2006) çalışmasında tanıtılan ters indeks (inverted indeks) kullanılarak yaprak düğümlerin ayırt edici özelliği, skorlamaya bir normalizasyon faktörü olarak eklenmektedir. Yerel özniteliklerin ayrıt ediciliğini arttırmak amacıyla Wang'ın çalışmasında (Wang, 2011), her özniteliğin katkısını adaptif olarak kontrol eden yeni bir ağırlıklandırma önerilmiştir. Adaptif kontrol edilerek, öznitelik ve uzamsal (spatial) yönteminin sıkılaştırılması önerilmiştir. Whan'ın çalışmasında yerel özniteliklerin bağlamsal ağırlıklandırılması ile sözcük ağacı kullanılarak getirme yaklaşımı güçlendirilmiştir (Wang, 2011). Thai'nin çalışmasında, sözcük ağacına entegre bir geometrik yeniden skorlama metodu sunulmakta ve geometrik skorlama hesaplama zamanını azaltılırken eşleme performansı arttırılmaktadır (Tsai, 2010). Zhang (Zhang, 2010), sözcük gruplama (Bag of words) yöntemindeki başarımı arttırmak için geometrik görüntü ifadelerini koruyacak şekilde yerel öznitelikleri uzamsal öğelere indirgemektedir.

Nister (Nister, 2006) ve Lowe'den (Lowe, 2004) esinlenerek, sözcük ağacındaki yaprak düğümlerin, daha üstteki düğümlere göre ayırt ediciliğinin fazla olması, bu tez çalışmasında eşleme başarımını arttırmak ve hesaplama karmaşıklığını azaltmak için yaprak düğümlere odaklanmamızı sağlamıştır. Yaprak düğümleri kullanarak ayırt ediciliği arttırmak için geniş ölçekli görüntü veritabanlarında arama ve getirme amacıyla bir eşleme skorlama yöntemi önerilmiştir. Görüntüye has ayrıt edici bağlamsal bilginin, sorgu ve veritabanı görüntüleri tarafından paylaşılan yaprak düğümlerde olduğu düşünülmektedir. Nister (Nister, 2006) gibi tüm düğümlerde veya Chen (Chen, 2010) gibi sorgu görüntüsünün ziyaret ettiği tüm düğümlerde, eşleme skorlamasını hesaplamak yerine, sorgu görüntüsünün sözcük ağacına yayılmış yaprak düğümlerinde kümülatif toplam hesaplayan yeni bir ağırlıklandırma formüle edilmiştir. Stanford'ın mobil görsel arama veri seti kullanılarak elde edilen sayısal sonuçlar, önerilen eşleme kriterinin eşleme performansını arttıran ve hesaplama karmaşıklığını azaltan yeni bir ayırt edici ölçüm yöntemi olduğunu göstermiştir.

Bu tez çalışması kapsamında ise bahsettiğimiz dört metod gerçeklenerek performans Stanford mobil görsel veri seti üzerinde denemiş ve sonuçlar karşılaştırmalı olarak raporlanmıştır. Yapılan çalışma kapsamında farklı mobil cihazlarla çekilmiş farklı kategorideki 100 referans görüntüsü için dört yöntemin de ilk 15 eşlemedeki başarımına bakılmıştır. Ortalama eşleme değerlerine bakılarak ilk eşleme başarımına bakıldığında sözcük ağacı ile uzaklık eşleme başarımı %43 iken ters indeks ile %45, bağlamsal ağırlıklandırma ile %47 ve önerilen yöntem ile %72 doğru eşleme oranına ulaşıldığı görülmektedir. İlk 5 eşlemeye bakıldığında, ters indeks ve önerilen arama/eşleme yönteminin %86, sözcük ağacının %66, bağlamsal ağırlıklandırmanın ise %76 başarım oranı olduğu görülmektedir. En son olarak ilk 15 eşleme başarımına bakıldığında ise tüm yöntemlerin %90, %99 seviyelerinde eşleme başarıma ulaştığı görülmektedir.

Sonuç olarak, farklı mobil cihazlardan elde edilen farklı kategorilerdeki görüntüler üzerinde yürütülen deneylerde, eşleme başarımı sonuçlarını karşılaştırmak için tüm cihazlar ve kategoriler için sabit olan girdiler tespit edilerek test koşulları oluşturulmuştur. Örneğin, kategorik bir karşılaştırma yapılması için dört farklı kategori için de ortak ve sabit olan Droid telefonu seçilmiştir. Droid en yüksek sonuca %94 başarımla kitap kapaklarında ulaşırken, en düşük başarımı tablolar üzerinde elde etmektedir. Bunun yanı sıra, iPhone cihazın tüm cihazlar arasında %98 sonuçla, en yüksek ilk seferde eşleme başarımına sahip olan telefon olduğu tespit edilmiştir.

IMAGE INDEXING AND MATCHING USING LOCAL FEATURES

SUMMARY

In this thesis, it is aimed to develop a mobile image search, matching and retrieval system which provides high accuracy while reducing the computational complexity. To achieve this various image retrieval and matching methods are evaluated and their performance are compared in order to determine effectiveness of different methodologies under different attacks.

Large-scale image retrieval based on visual features has been a challenging research topic because of many emerging applications specifically the mobile image search. Most of the scalable image retrieval algorithms fall into two categories. They either employ discriminative floating-point local features indexed by a vocabulary tree (Nister, 2006) or holistic features indexed by compact hashing codes (Torralba, 2008). Methods to integrate these techniques are also introduced in Zhang's study (Zhang, 2015). In this thesis, we propose a visual image retrieval algorithm for mobile search. These type of applications the query is just a local pattern located in the image captured by the mobile device (Chandrasekhar, 2011). Size of the local pattern, its location and background pattern may vary from image to image depending on the type of mobile sensor and the environment. We describe a SIFT descriptor based vocabulary tree indexing method which is powerful in identifying near-duplicate images or regions since local features are particularly capable of representing regional image patterns.

In the literature there are several methods employing vocabulary tree based indexing and retrieval. The method proposed as scalable recognition with a vocabulary tree in Nister's study (Nister, 2006) can be considered as the pioneering work in vocabulary tree based image matching. Inspired by this study, the vocabulary tree based approach has been further improved in Philbin's study (Philbin, 2007) that employs RANSAC as a post spatial verification to check whether the SIFT features of retrieved images to have a similar layout to the query. In Chen's study (Chan, 2010), a compression scheme for the inverted index introduced in Nister's study (Niser, 2006) is proposed to minimize the memory requirements of a vocabulary tree based indexing and to speed up the retrieval. In Chen's study (Chan, 2010) also the discriminative property of the leaf nodes is used by including them into the matching score as a normalization factor. In order to enhance the discriminative power of individual local features, a new weighting score which adaptively controls the contribution of each descriptor is proposed in Wang's study (Wang, 2011). The vocabulary tree-based retrieval approach is improved by adaptively controlling the contextual weighting of in individual features in both descriptor and spatial domains (Wang, 2011). In Tsai's study (Tsai, 2010), a geometric re-ranking method that utilizes a vocabulary tree integrated with a location geometric scoring method is presented and it is shown that the geometric scoring reduces the computation time while increasing the recognition performance. Zhang et al. (Zhang, 2010) quantizes spatial offsets among local features through the geometry-preserving visual phrases to improve the accuracy of BoW (Bag of words).

Inspiring from Nister (Nister, 2006) and Lowe (Lowe, 2004) which clearly state that the leaf nodes are more distinctive than the upper level nodes of vocabulary tree, we also focus on the leaf nodes in our matching scheme, to increase accuracy while reducing the computational complexity. In order to incorporate the leaf nodes

effectively for search and retrieval on large-scale image databases we propose a novel matching score to boost their discriminative ability. We believe in that the discriminative contextual information pertinent to individual images can be reached at the leaf nodes shared by the query and database images. Instead of calculating the matching score over all nodes of the vocabulary tree as Nister (Nister, 2010) or over all nodes visited by the query image as Chen (Chen, 2010), we formulate a novel weighting score calculated as cumulative sum of the contributions of the leaf nodes of the query image spread throughout the vocabulary tree. Numerical results obtained on Stanford's Mobile Visual Search data set demonstrate that the proposed matching criteria can be used as a new discriminative measure to improve the visual matching performance while reducing the retrieval complexity.

Big data concept is a concept that traditional data processing is not sufficient for large and complex data sets. One of the most challenging part of the big data concept is that the data is mostly unstructured. Unstructured data is not created through a defined pattern and can include complex data sets in itself. We can see that the same situation can occur for visual image search large data sets. For each image there are different number of SIFT features. Then, the size of the value is different and complexity can be changeable.

In our implementation, we used hierarchical K-means clustering to perform BoW on SIFT descriptors extracted from the database images for building VT indexing of individual categories. We used 'db4o' which is an open source Java based object oriented database. The idea behind using an object oriented database rather than a relational or XML based database is db4o enables us creating nested objects i.e., cluster (node) object and keypoint (descriptor) object.

The developed system consists of a number of subsystems. First the local keypoints of reference images are extracted and feature vectors are created by SIFT in the development of mobile image matching system. Afterwards, vocabulary tree is created applying hierarchical k-means clustering on the local feature vectors corresponding to local key points. This vocabulary tree structure is saved in order to serve as a learning set during matching process. Training images including CD, DVD and Book covers and art paintings are gathered from Stanford mobile visual dataset.

After construction of the reference database, the system extracts features of the query image and starts to search them in the vocabulary tree in order to find the proper match with reference images stored in the database. The matchingis declared between the reference and query images which have the most similar features.

During the experiments, we have tested and evaluated different methods we have applied in this work. The first method is based-on determining similarity scores for query images by calculating Euclidian distance between features, which are indexed and weighted (Nister, 2006). The second method, which is based-on entropy in vocabulary tree using inverted index (Girod, 2011). The third method is the implementation of contextual weighting approach for mobile image matching (Wang, 2011). The final method is the proposed approach in this thesis, which suggests a different weighting technique based on the leaf nodes of the vocabulary tree.

Performances of the image matching of mobile devices are dependent on resolution, density, position and the complexity of the reference image. In order to prevent this, we applied sequenced iterative sum approach during geometric verification. Furthermore, we have assigned weights for each feature, which are indexed on final

leaf nodes of the vocabulary tree. The threshold score is calculated by summing each value for every reference.

We have evaluated four different approaches, which are presented in this thesis using Stanford Mobile Visual Dataset. A hundred reference images in different categories and captured from different mobile devices are used for training and testing of four models. We have compared four different approaches by collecting matching results of first 15 attempts. According to the experiment results, it is shown that proposed approach increase the accuracy of the mobile visual search and matching system besides vocabulary tree, the inverted indexing and contextual weighting methods. Especially, the outcome of the fourth case shows that the new proposed approach significantly improves the search performance and increases the matching accuracy . In the first case the average matching rate is 43% whereas it is reported as 45% for the second test case. The third case which applies the contextual weighting has %47 matching rate. On the other hand the fourth and the proposed search/retrieval approach has 72% average succefully matching rate. For the first 5 attempts, inverted indeks and the proposed search/ retrieval approach has %86, vocabulary tree based approach has % 66 and contextual weighting has %76 success rates. At last we can see that all the methods have %90 - %99 sucess rates for the first 15 attempts.

In conclusion, the experiments are performed on different image categories captured by different mobile devices by creating test cases after determining constant inputs for all devices and categories. For example, in order to make categorical comparison, Droid is selected because of the fact that is a common device for all four catogories. It achieves the highest accuracy on book covers which is 94% success rate and the lowest rate is reported on paintings which is 56%. Moreover, iPhone achieved the highest accuracy with 98% success rate on the first attempt during the comparison of the success rates of all devices.

1. GİRİŞ

Görsel olarak birbirine benzeyen görüntülerin geniş veri tabanlarından etkili bir biçimde eşlenmesi ve getirilmesi problemi, bilgisayarla görü alanında her geçen gün daha fazla araştırmanın konusu olmaktadır (Wang ve diğ., 2011). İnternet kullanımının yaygınlaşmasıyla birlikte, dijital görüntülerin kullanımında çok büyük artış yaşanması ve bu artışın küresel ölçekteki etkileri bu alandaki çalışmalara olan ilginin önemli sebeplerinden biridir (Junior, Marçal ve Batista, 2014).

Günümüzde mobil cihazların kullanımındaki büyüme ise görsel arama ve eşleme çalışmaları üzerinde farklı ihtiyaçları beraberinde getirmiştir. Mobil cihazların gelişmiş kamera uygulamalarıyla kullanıcılara görüntü yakalama, kaydetme, işleme ve paylaşma gibi yetenekler sunması, kullanıcılara görüntü tabanlı bilgi edinme ve bilgi paylaşma olanağı vermektedir. Bununla birlikte; ürün, film, cd, poster, sanat eserleri hakkında, görsellerini kullanarak aramaya, bilgi almaya veya karşılaştırmaya imkân tanıyan uygulamaların geliştirilmesi mobil kullanıcıların talepleri ve beklentilerine örnek gösterilebilir. Diğer taraftan bu talepler mobil görüntü eşleme teknolojilerinin daha da önem kazanmasını beraberinde getirmektedir (Girod, 2011). Bu alanda şimdiye kadar yapılmış çalışmalara örnek olarak Google Goggles (Hopkins, 2015) , Nokia İşaret Et ve Bul (CNET, 2009), SnapTell (Rao, 2008) ve Kooaba (Kooaba.com, 2015) gibi uygulamalar gösterilebilir.

Mobil görüntü arama, getirme ve eşleme teknolojilerinin gün geçtikçe önem kazanmasının yanı sıra bu alanda yapılan çalışmalarda özel bir takım zorluklarla karşılaşılmaktadır. Bu zorlukların bazıları dijital kopyaların (bu çalışmada mobil cihazlarla çekilmiş görüntüler) orijinal görüntülere göre farklı ışıklılık koşullarında çekilmiş olması, bulanıklık, orijinal ile çekilen boyut farklılığı, kırpılma gibi sebeplerden kaynaklanan bozulmalar olarak sıralanabilir (Wang ve diğ. 2011). Bu faktörler görüntü eşleme performansını doğrudan etkileyebilirler. Bunun yanı sıra genel olarak büyük veri işleme probleminin bir parçası olan; görüntülerin büyük veri tabanlarından hızlı ve başarılı bir biçimde getirilmesi problemi bir diğer zorluktur. Mobil cihazların sınırlı hafıza ve işlemci kabiliyetine sahip olmaları göz önüne

alındığında bu problem ciddi önem taşımaktadır. Veri tabanlarındaki bilginin büyük boyutlu olmasının yanı sıra, arama ve eşlemenin yapılacağı sunuculara erişimde iletişim gecikmeleri tüm işlem süresinin önemli bir kısmını oluşturmaktadır. Bu durum görüntü eşleme algoritmalarının yüksek doğrulukla ve hızlı çalışması gereksinimini daha da önemli kılmaktadır.

Geniş ve büyük ölçekli milyonlarca görüntü arama/eşleme çalışmalarında özellikle ana iki yaklaşım ön plana çıkmaktadır. Terminolojiye baktığımızda bu yaklaşımlardan bir tanesi yerel özniteliklerin kullanıldığı yöntemler, diğeri bütünsel (holistic) özniteliklerin kullanıldığı yöntemlerdir (Zhang, 2015). Yerel öznitelikler kullanılarak yapılan çalışmalarda daha detay arama ve eşleme yöntemleri kullanılabilmekte fakat eşleme zamanı ve karmaşıklığı artmaktadır. Örneğin farklı bakış açılarından çekilen görüntüler ayırt edilebilmektedir. Bütünsel öznitelikler kullanılarak yapılan çalışmalarda ise global öznitelikler kullanıldığı için döndürme, ışıklılık değişimi gibi bilgiler bulunmamakta fakat öznitelikler daha az yer kaplayan veriler olduğu için orijinal görüntülerden hızlı arama ve eşleme sağlanabilmektedir (Zhang, 2015).

Yerel ve bütünsel öznitelikler kullanılarak yapılan çalışmalarda her bir yöntemin kendine özgü artı ve eksileri olduğu görülmektedir. Şekil 1.1 de bütünsel öznitelikler kullanılarak daha başarılı eşlemelerin yapıldığı örnekler görülmektedir. Solda sorgu görüntüsünü, üst sırada bütünsel öznitelikler kullanılarak eşlenen referans görüntülerini, alt sırada yerel öznitelikler kullanılarak yapılan eşleme sonuçlarını göstermektedir. Özellikle bütünsel öznitelikler görüntünün tümünü algılamada ve betimlemede, renk histogramları ve gist öznitelikleri sayesinde aynı pozisyonda veya renk benzerliğindeki görüntüleri eşlemede daha başarılı görülmektedir.

Bütünsel öznitelikler kullanılarak yapılan çalışmalarda, yerel özniteliklerin tümünün eşleşmesi beklenmediği için sorgu görüntüsünün farklı yönelim, ışıklandırma, ölçek, rotasyon gibi varyasyonları için ayrı referans görüntülerine gereksinim yoktur. Bu yüzden bu yöntem özellikle dağınık arka plan söz konusu olduğunda çok daha gürbüzdür. Şekil 1.1 de arka plan görüntülerinde bu durum açıkça görülmekte ve eşleme başarımını etkilemektedir.



Şekil 1.1 : Bütünsel (holistic) öznitelik ile yapılan eşleme (Zhang, 2015).

Şekil 1.2 de yerel öznitelikler kullanılarak daha başarılı eşlemelerin yapıldığı örnekler görülmektedir. Solda sorgu görüntüsünü, üst sırada bütünsel öznitelikler kullanılarak eşlenen referans görüntülerini, alt sırada ise yerel öznitelikler kullanılarak yapılan eşleme sonuçları görülmektedir. Yerel özniteliklerin sözcük ağacı kullanılarak yaptığı eşlemeler Şekil 1.2 den de görüleceği gibi komşu özniteliklerle olan ilişki özelliği, renkten bağımsız nesnelerin dönme, ışık, ölçek ve yer bilgilerini de içermesi sebebi ile yerel öznitelikler sorgu görüntüsünün bir parçası aranıyor ise daha gürbüzdürler. İki yöntemde farklı kullanım alanlarına göre tercih edilebilir durumda olduğu gibi örnekleri Şekil 1.1 ve Şekil 1.2 de görülmektedir. Tez çalışması kapsamında sözcük ağacı kullanarak yerel öznitelikler üzerinde arama/eşleme çalışmalarına katkıda bulunulmuştur.



Şekil 1.2 : Yerel (local) öznitelikler ile yapılan eşleme (Zhang, 2015).

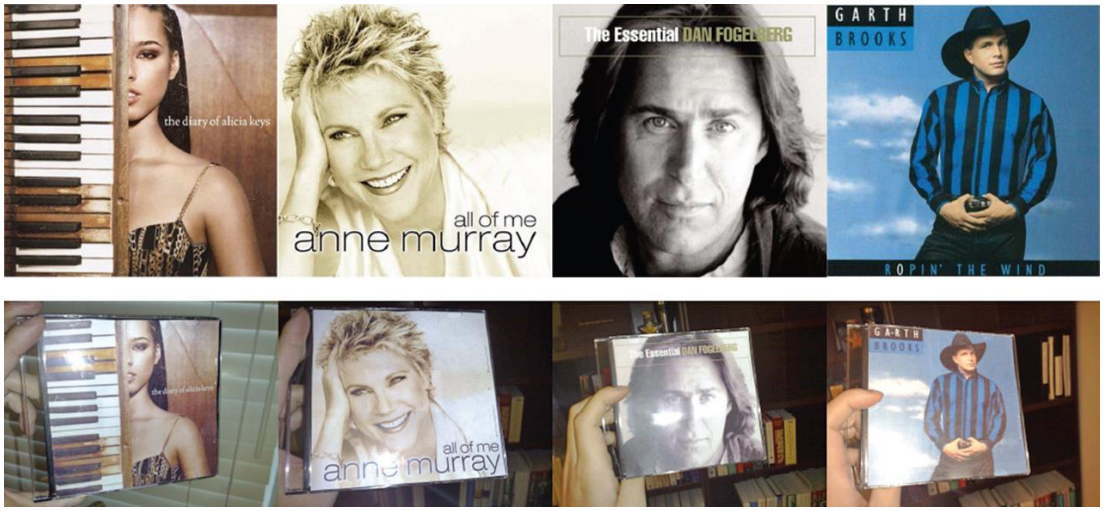
Yerel öznitelik tabanlı yöntemlerin en önemli avantajları aşağıda sıralanmaktadır:

- İlgili nesnelerin belirli kısımlarının görünmediği durumlarda da başarılı tanıma yapılmaktadır (Şekil 1.2).
- Bilinen nesnelere bağlı olarak model yapılandırılması, bir diğer ifadeyle öğrenme, otomatik olarak gerçekleştirilir. Eğitim için referansların seçilmesi dışında herhangi bir kullanıcı müdahalesi gerekli değildir.
- Yerel temsil görünüme dayalıdır. Herhangi bir geometrik çıkarım (çizgi, daire vb.) veya tasvire ihtiyaç yoktur.
- Tanıma için nesnelerin arka plandan segmentasyonuna ihtiyaç bulunmamaktadır.
- Hem veri tabanı hem de sorgu görüntülerinde yapılan hesaplamalar birebir aynı şekilde gerçekleştirilmektedir.

Yerel özniteliklere dayalı yaklaşım nesne ışıklılık değişimleri, örtüşme, poz değişimleri ve karmaşık arka plan durumlarına karşı gürbüzdür. Bir görüntüdeki nesnenin bir bölümünün diğer bir nesne tarafından kapatıldığı durumda örtüldüğü kısma karşı düşen yerel özniteliklerde kısmi kayıp yaşanabilmektedir. Fakat yeterli sayıda öznitelik hesaplanabildiğinde, örtüşme nesne tanımlamada sorun yaratmamaktadır. Karmaşık arka plan (cluttered background) problemi de tanıma / eşleme sürecinin son aşamasında çözümlenmektedir. Çünkü varsayımsal eşleşmelerin tümü son aşama olan doğrulama (verification) aşamasında gözden geçirilerek onaylanır ve yanlış benzeşmeler elenir.

Bu tez çalışması kapsamında, ilgilenilen problemde yerel öznitelikler kullanıldığı için Şekil 1.3'deki gibi mobil sorgu ve referans görüntüler ile mobil görüntü eşleme üzerine yapılan önceki çalışmalar ve yaklaşımlar araştırılarak daha hızlı ve daha doğru eşleme kabiliyetine sahip bir sistem tasarlanmıştır. Sistemin daha önce belirtilen ölçeklendirme, ışıklılık değişimi, bulanıklık, döndürme, kırpılma gibi bozulmalara karşı gürbüz olması için SIFT (Scale-Invariant Feature Transform) öznitelikleri kullanılmıştır (Löwe, 2004). İlgilenilen probleme yerel öznitelikler kullanıldığı için

Şekil 1.3'den de görüleceği gibi mobil cihazlar ile çekilmiş Farklı indeksleme ve ağırlıklandırma metotlarının eşleme başarımındaki etkilerini gözlemlemek için ise sözcük ağacı (vocabulary tree) üzerinde ters indeks (inverted index) ve bağlamsal ağırlıklandırma (contextual weigting) yöntemleri uygulanarak performans karşılaştırması yapılmıştır (Wang ve diğ., 2011). Eşleme başarım testlerinde Şekil 1.3'de örnekleri bulunan Stanford Üniversitesi tarafından mobil uygulama araştırmaları için oluşturulmuş Stanford Mobile Visual Search Data Set (Chandrasekhar ve diğ., 2011) veri tabanı kullanılmıştır.



Şekil 1.3 : Mobile Visual Search Data Set örnekleri (Chandrasekhar, 2011).

Performans karşılaştırmalarında, sözcük ağacı (Nister ve diğ., 2006), ters indeks (Girod ve diğ., 2011), bağlamsal ağırlıklandırma (Wang ve diğ., 2011) ve bu tez çalışması kapsamında yaprak düğümlere dayalı yeni bir arama/eşleme yöntemi kullanılmıştır. Tüm bu yöntemler db4o veri tabanı ortamında gerçek uygulamalardakine benzer şekilde kullanılarak bir mobil görüntü arama/eşlemesi sistemi geliştirilmiş ve karşılaştırmada kullanılan bu yaklaşımların eşleme başarımına etkilerini gözlemlenmiştir.

Mobil görüntü arama uygulamalarında görüntü eşleme amacıyla kullanılan referans görüntülere ilişkin özniteliklerin tamamının veri tabanında saklanması ve sorgu görüntüsüne ilişkin özniteliklerin veri tabanında kayıtlı bulunan bütün referans özniteliklerle karşılaştırılması şeklinde gerçeklenen kapsamlı arama (exhaustive search), işlemsel yükün çok fazla olması nedeniyle, kullanışlı değildir. Bu amaçla geliştirilen çözümlerin en çok kabul görenlerinden birisi özniteliklerin saklanmasında sözcük ağaçları (vocabulary tree) kullanılmasıdır. Sözcük ağaçları temel olarak

içerisinde bulunan sözcükleri birbirlerine benzerliğine göre hiyerarşik bir biçimde gruplayıp saklama temeli üzerine kurulur.

Sözcük ağacı kullanarak büyük ölçekli görüntü tanıma ve eşleme için birçok yöntem kullanılmıştır. Bu yöntemlerin çoğu öznitelik tabanlı arama ve eşleme şeklindedir. SIFT tabanlı yaklaşımların en popüler olanı sözcük ağacı ile ölçeklenebilir eşleme (Nister, 2006) çalışmasıdır. Bu çalışma Stanford mobil veri seti kullanılarak da uygulanmıştır (Çırakman, 2012). Çalışmada referans görüntülerden elde edilen öznitelikler üzerinde hiyerarşik k-ortalamalı kümeleme kullanılarak oluşturulmuş sözcük ağacı yapısı kullanılmıştır. Bunun en önemli sebeplerinden biri ise, sözcük ağacı yapısı üzerinde arama-getirme işlemini hızlandıracak ağırlıklandırma metodunun kolaylıkla uygulanabilmesidir.

Niseter'in çalışmasında sözcük ağacı içerisindeki her bir düğüm için bir entropi değeri hesaplanmaktadır. Bu entropi değeri, ağaçta toplam referans görüntüsü sayısı ile o düğümden geçen referans görüntü sayısının logaritmik oranıdır. Sorgu görüntüsünün eşlemesi sırasında eşleme yapılmadan önce sorgu görüntüsünün öznitelikleri ağaca, düğümlerin entropi değerlerine bakılarak Öklid uzaklığı (Euclidean distance) ile yerleştirilmektedir. Eşleme sırasında ise her bir düğümdeki entropi değeri ile sorgu ve referans görüntülerinin öznitelik sayıları çarpılarak bir ağırlık değeri bulunmakta ve bu ağrılık değerleri arasında Öklid uzaklığı hesaplanarak en yakın referans görüntüsü eşlenmeye çalışılmaktadır.

Bu yönteme ek olarak ters indeks yapısı kullanılarak arama/eşleme başarımı arttırılmaya çalışılmıştır. Ters indeks (inverted index) veri yapısı özellikle arama motorlarında sıkça kullanılan bir tekniktir. Büyük veri tabanlarında hızlı görüntü eşleme gerçekleştirmek için, sözcük ağacı veri tabanında bulunan her ağaç düğümünü ziyaret eden görüntüleri ve her düğümdeki görüntü sayılarını eşleyen bir ters indeks dizini tutulmaktadır (Girod, 2011).

Özellikle halı, çim vb. benzer renk ve nesneleri içeren görüntülerdeki eşleme başarımını artırmak için, özniteliklerin sözcük ağacı boyunca izlediği yol (path) bilgisini de hesaba katan bağlamsal ağırlıklandırma yöntemi önerilmiştir (Wang, 2011). Özniteliklerin ayrıştırıcı gücü, özdeş olanlar dahil, görüntüden görüntüye değişmektedir. Birbirine benzer fakat çok sayıda özniteliğin olduğu durumlarda özniteliklerin ağacın yaprak düğümlerine doğru izledikleri yolların farklılaştığı

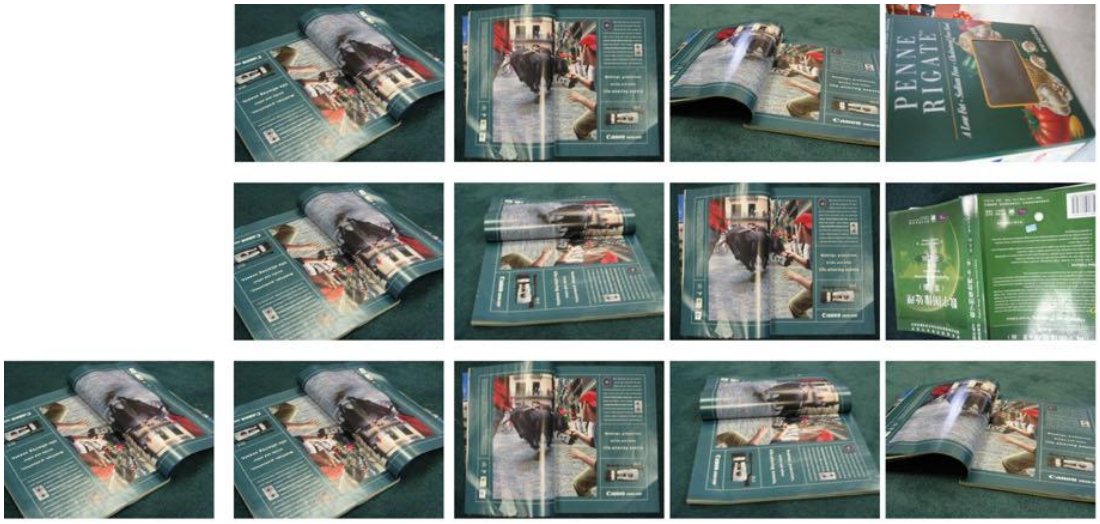
gözlemlenmiştir. Bu yol bilgisi ayrıştırıcı bir özellik olarak kullanılarak görüntü eşleme gerçeklenmiştir. Sorgu görüntüsü özniteliklerinin her birinin izlediği yol boyunca geçtiği düğümlerde hem sorgu hem de aynı yolu izleyen referans görüntü öznitelikleri için birer ağırlıklandırma değeri hesaplanmıştı. Bu ağırlıklandırma değerleri o düğüme ait entropi değeri ile beraber hesaplanarak o düğüme ve o referans görüntüye ait yeni bir ağrılık ve skor değeri oluşturulmuştur. Bu skor değerleri referans görüntü için toplam skorlama değeri elde edilmiştir. Çalışma kapsamında ayrıca uzamsal bağlamsal ağırlıklandırma (spatial contextual weightning) adı verilen, SIFT özniteliklerindeki ölçekleme, rotasyon ve ışık etkisi gibi özelliklerini de birer ağırlık değeri olarak hesaba katan bir ağırlıklandırma yöntemi önerilmiştir.

Yanlış eşlenen öznitelikleri elimine etmek için literatürde son dönemde eşleme sonrası geometrik sağlama yöntemi uygulanmaktadır (Tsai, 2010). Geometrik sağlama yöntemi, RANSAC (random sample consensus) şeklinde iteratif bir yöntem kullanarak hem sorgu hem de referans görüntülerin özniteliklerinin karşı düştüğü lokasyonlarının tespitini yapabilmektedir. Önerilen yöntemde sorgu görüntüsünün, referans görüntüler ile belli bir skorlamaya göre eşlenmesi sonrası, en benzer referans görüntüler ile sorgu görüntüsünün özniteliklerinin geçtikleri ortak düğümler çıkarılır. Ortak düğümler üzerinde, sorgu ve referans görüntü özniteliklerinin yönelim (orientation) ve ölçek (scale) değerleri de kullanılarak yeni bir skorlama yapılır. Önerilen yöntemlere bakıldığında özellikle milyonlarca referans görüntü arasından sorgu görüntüsünün eşlenme süresi, hesaplama karmaşıklığının artması sonucu, çok zaman almaktadır. RANSAC vb. yöntemlerin optimize edilerek eşleme süresinin dramatik şekilde azaldığı çalışmalar da mevcuttur. Geometrik sağlama yöntemine, eşleme sonrası eşlenen sorgu ve referans özniteliklerinin bir alt kümesi alınarak RANSAC yerine yukarıda bahsi geçen uzamsal bağlamsal ağırlıklandırma yöntemi uygulanarak eşleme süresi kısaltılabilmektedir (Wang, 2011).

Bütünsel ve yerel özniteliklerin avantaj ve dezavantajlarına bağlı olarak eşleme ve skorlama sağlamayı amaçlayan yöntemler de mevcuttur. Zhang'ın (Zhang, 2015) önerdiği yöntemde her iki metotla da elde edilen skorlar yeniden sıralanmaktadır. Bu sıralama için oluşan skorlama değerlerinin ortak bir şekilde tekrardan sıralanması gerekmektedir. Bu sıralama için en önemli zorluk sonuçların metotlara göre kalitesi ve başarımının herhangi bir yöntem veya yönlendirici bir girdi olmadan karar

verilmesidir. Böylece çalışmadaki en önemli fikir, sorguya özel bir kalitenin belirlenerek üst seviye çıkan referans görüntülerin sıralanmasıdır.

Şekil 1.3 de sol taraftaki görüntü sorgu görüntüsüdür. En üst satırdaki görüntüler bütünsel (holistic) öznitelikler kullanılarak en yüksek skor ile eşlenenden sırasıyla başlayacak şekilde yapılan ilk 4 eşlemenin sonucunda veri tabanından getirilen görüntüleri göstermektedir. 2. satırda yerel öznitelikler kullanılarak elde edilen sonuçlar görünmektedir. En alt satırda ise önerilen hibrid yöntem ile elde edilen ilk 4 eşleme sonuçları gösterilmektedir. Hibrid eşlemenin 4. eşlenen görüntüde yapılan yanlış eşlemeyi önlediği görülmektedir.



Şekil 1.4 : Üst satır bütünsel, ikici satır yerel ve birlikte kullanımı (Zhang, 2015).

Bu tez çalışması kapsamında önerilen arama/eşleme yöntemi, sözcük ağacı üzerinde SIFT yerel öznitelikler kullanılarak, yaprak düğümlerde sorgu ve referans görüntüler için birer ağırlık değeri oluşturmakta ve bu ağırlık değerlerini iteratif bir şekilde hesaplayarak veri tabanındaki her bir referans görüntüsü için bir skorlama değeri hesaplanmaktadır. Veri tabanı oluşturulurken ağaca eklenecek sözcükler k-ortalamalı öbekleme ile belirlenmektedir ve her bir referans görüntüsünün öznitelikleri ile birlikte ağaca yerleştirilmektedir. Bu yerleştirme k-ortalamalı öbekleme ile oluşturulmakta ve her bir referans görüntüsünün öznitelikleri ağaca yerleştirilmektedir. Bu yerleştirme ve veri tabanı oluşturma aşamasında yukarıda bahsi geçen sözcük ağacı uzaklığı (Nister, 2006; Çırakman, 2012) ve ters indeks (Girod, 2011) yöntemlerindeki entropi değeri hesaplama ve referans sayısı uzunluğundaki vektörler de bu değerleri

indekslemede kullanılmıştır. Ayrıca nesne tabanlı veri tabanı kullanılarak, öznitelik, düğüm, görüntü gibi nesneler oluşturulmuş ve birbirleri ile ilişkilendirilmiştir. Örneğin ağaç yapısında hangi düğümden hangi referans görüntüsünün kaç özniteliği geçmektedir veya referans görüntüsü hangi düğümlerden kaç kez geçmektedir gibi bilgiler ilişkilendirilebilmektedir. Entropi ağırlık değerleri ağaç oluşturulurken bir kez hesaplanmakta ve veri tabanında saklanmaktadır.

Tez çalışması kapsamında başarım testlerinde kullanılan Stanford mobil görüntü veri seti içerisindeki CD, DVD, kitap ve sanatsal tabloların her biri için farklı veri tabanı oluşturulmuştur. Bu veri tabanları referans görüntülerin farklılığından dolayı farklı sayıda öznitelik ve buna bağlı olarak farklı sayıda seviyelere ve farklı sayıda yaprak düğümlere sahiptir.

Sorgu görüntüsü öznitelikleri sözcük ağacına Öklid uzaklığı en küçük olan düğümler izlenerek yerleştirilmektedir. Bu yerleştirme sonrası sırasıyla yerleştirildiği her bir yaprak düğüm için sorgu ve referans görüntüye ait birer ağırlık değeri hesaplanmaktadır. Bu ağırlık değerleri ve o düğüme ait entropi değeri çağrılarak referans görüntüye ait bir skorlama değeri oluşturulmaktadır. Bu değer tüm ortak yaprak düğümler için hesaplandıktan sonra toplanarak referans görüntü için bir toplam skorlama değeri elde edilmektedir. Tüm referans görüntüler için oluşan skorlama değerleri sıralanarak en büyük olanı en iyi eşlenen görüntü olarak raporlanmaktadır.

Bu tez çalışmasında, Bölüm 2'de SIFT yerel öznitelikleri hakkında bilgiler verilmiş ve görüntü içerik modelleme kavramı üzerinden durulmuştur. Bölüm 3'de, mobil görüntü içerikleri kullanılarak sözcük ağacı oluşturma yaklaşımından bahsedilmiştir. Aynı bölümde literatürde sözcük ağacı kullanarak farklı indeksleme ve ağırlıklandırma metotlarının eşleme başarımındaki etkilerini gözlemlemek için sözcük ağacı (Nister, 2006), ters indeks (Chen, 2010) ve bağlamsal ağırlıklandırma (Wang, 2011) yöntemleri ile bu tez çalışması kapsamında önerilen yaprak düğüm bazlı arama eşleme yöntemi anlatılmıştır. Bölüm 4'de, dört yaklaşımın Stanford mobil görüntü veri seti kullanılarak bulunan eşleme başarım sonuçları karşılaştırılmış ve yorumlanmıştır. En son olarak da sonuçlar kısmı olan Bölüm 5'de tez çalışması kapsamında ortaya çıkan bulguların kısa bir özeti ve gelecek araştırmalar için öneriler paylaşılmıştır.

2. YEREL ÖZNİTELİKLERLE GÖRÜNTÜ İÇERİK MODELLEME

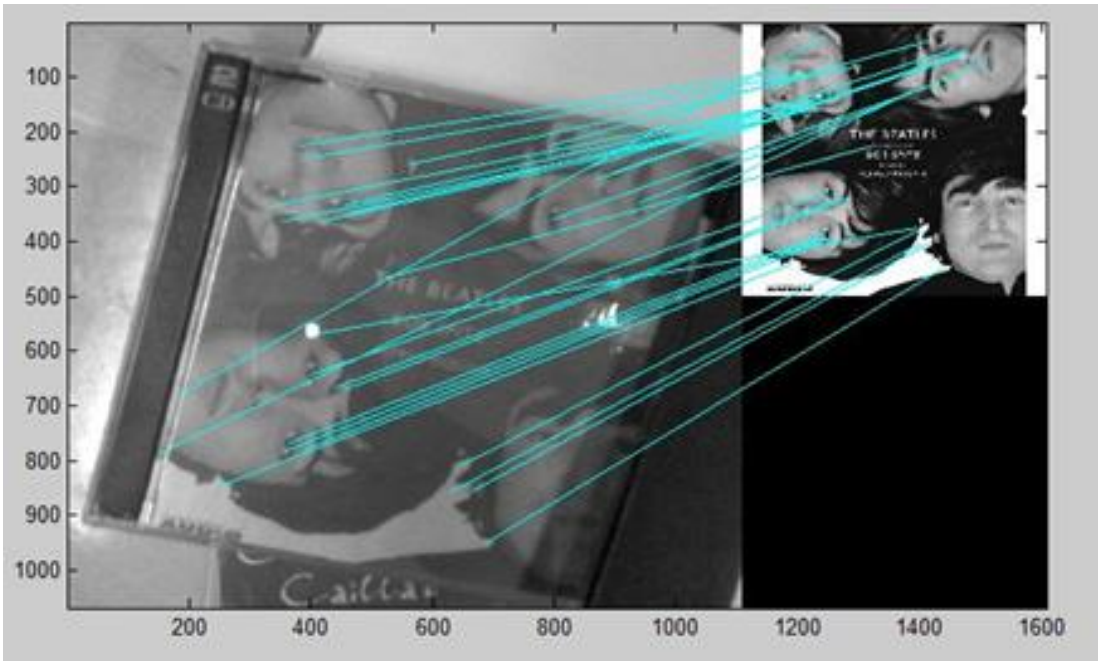
Bir görüntüyü diğer bir görüntü ile eşlemek amacıyla kullanılan bir çok yerel öznitelik vardır. Bunlardan en popüler olanları SIFT (Scale – Invariant Feature Transform), SURF (Speeded Up Robust Features) ve FAST (Fast Affine Template Matching) özniteliklerdir. Bu özniteliklerin çıkarımında işlemsel karmaşıklık dolayısıyla hız farklıdır ve eşlemede kullanıldıklarında elde edilen başarım da farklıdır. SURF ve FAST öznitelikleri daha hızlı çıkarılırken eşleme başarımı açısından eşleme başarımı açısından SIFT öznitelikleri daha detay bilgi verdiklerinden daha başarılıdır.

Bu bölümde, mobil cihazlardan elde edilen görüntülerle gerçekleştirilen arama/eşleme uygulamasında kullanılan SIFT öznitelik çıkarma algoritması ayrıntılı şekilde ele alınmaktadır. Çalışmada, SIFT özniteliklerinin çıkarılması sırasıyla, ölçek uzayı uçdeğer kestirimi, anahtar noktaların konumlandırılması, anahtar noktalara bağımsız yönelimler atanması ve anahtar nokta tanımlayıcılarının (öznitelik vektörleri) yaratılması adımları izlenerek gerçekleştirilmektedir.

Ayrıca SIFT özniteliklerinin ölçek, ışıklılık değişimi, boyut değişimi ve rotasyon gibi koşullardan bağımsız oluşu bu bölümde incelenerek, literatür çalışmasında yer verilen yerel öznitelik yaklaşımın nesne tanıma problemi için diğer yaklaşımlara göre daha güçlü sonuçlar elde ettiği tespiti detaylandırılmaktadır.

SIFT (Scale – Invariant Feature Transform) öznitelik çıkarma algoritması (Lowe, 2004) görüntü işleme alanında görüntü yerel özellikleri modelleme amacıyla yaygın olarak kullanılan bir algoritmadır.

SIFT, geniş ölçüde iki boyutlu afin (affine) deformasyon, aşırı gürültülülük ve aydınlatma değişimleri gibi sorunlar söz konusu olduğunda da gürbüz eşleme yapabilme olanağı sağlar. SIFT öznitelikleri yüksek derecede ayırtedici yapıda olmaları sebebiyle, geniş bir veri tabanı üzerinde eşleme yaparken dahi doğruluk oranı yüksek eşleme başarımı sağlar. Aynı resmin farklı koşullar altındaki görüntülerinin SIFT kullanılarak eşlenmesine bir örnek Şekil 2.1 de gösterilmiştir.



Şekil 2.1 : CD kapağı SIFT öznitelik eşleme örneği.

2.1 Görüntü Özniteliklerinin Elde Edilmesi

Bir SIFT öznitelik vektörünü $f = \{s, x, y, \sigma, \theta\}$ şeklinde, s öznitelik tanımlayıcı, x, y yer bilgisi (location), σ ölçekleme ve θ yönelim (orientation) olarak tanımlayabiliriz.

SIFT öznitelikleri aşağıdaki 4 adım gerçeklenerek çıkarılır:

- Ölçek uzayı uçdeğer kestirimi
- Anahtar noktaların konumlandırılması
- Anahtar noktalara bağımsız yönelimler atanması
- Anahtar nokta tanımlayıcılarının (öznitelik vektörleri) yaratılması

2.1.1 Ölçek uzayı uçdeğer kestirimi

İlk aşama olarak görüntüde aranacak bir nesnenin farklı öncelikle, aynı nesnenin farklı görünümlerine atanabilecek bütün ölçekler ve görüntü adresleri araştırılır. Bu amaçla olası bütün görüntünün olası bütün ölçekler üzerindeki kararlı öznitelikleri araştırılır (Witkin, 1984). Koendrink (Koendrink, 1984) ve Lindeberg'in (Lindeberg, 1994) çalışmaları kullanılabilecek bir ölçek uzayı çekirdeğinin tek ölçek uzayı çekirdeğinin

Gaussian fonksiyonu olduğunu göstermişlerdir. Bu yüzden görüntünün ölçek uzayı, L(x, y, σ), görüntünün, I(x,y), (2.1)'de görüleceği gibi Gauss operatörü ile konvolusyonun, G(x,y,σ), sonucu olacak şekilde ifade edilmiştir.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.1}$$

Ölçek uzay uçdeğer kestirimi (2.1) formülasyonunda x ve y koordinatları, σ değeri ise ölçek parametresini temsil eder ve (2.1) formülasyonundaki Gauss operatörü alçak geçiren filtre görevi görür. Görüntünün Gauss operatörü ile konvolüsyonunun sonucunda bulanıklaşmış görüntü elde edilir. Bulanıklaştırılmış görüntü orijinalinden çıkarıldığında, ani gri düzeyi değişimlerinin olduğu noktaları belirginleştiren bir yüksek geçiren filtre gibi sonuç verir. Gauss operatörü (2.2)'de formüle edilmiştir.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2.2}$$

Ölçek uzayında kararlı anahtar nokta adreslerini sezinleyebilmek için, ölçek-uzay uçdeğerleri, DoG (Difference-of-Gaussian) fonksiyonu, D(x,y,σ), ile en yakın iki ölçek arasındaki uzaklık (2.3) formülasyonu ile hesaplanmaktadır.

$$D(x, y, \sigma) = \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y) \;\; = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.3}$$

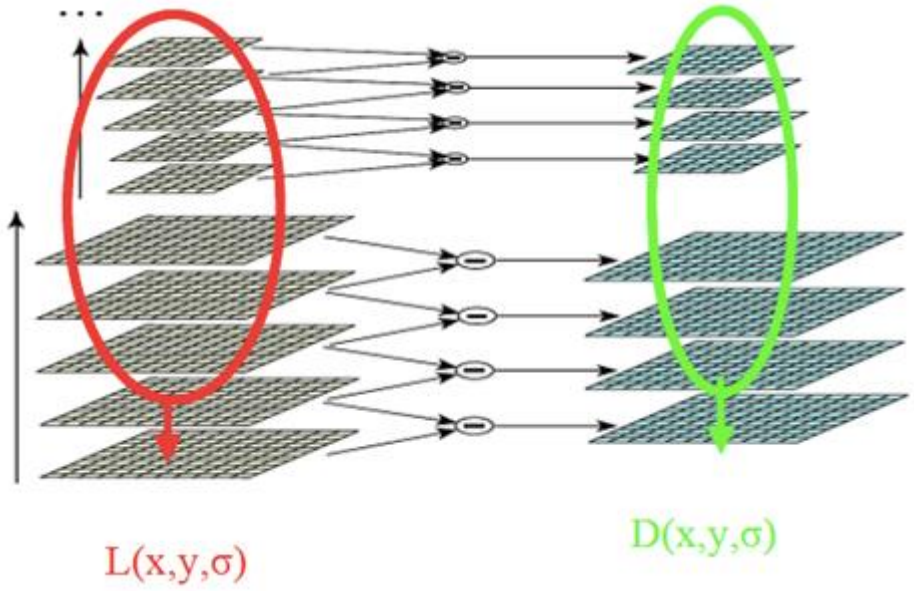
DoG fonksiyonu, ölçek-normalize Laplace-Gaussian, $\sigma^2\nabla^2$G, metoduna oldukça benzer bir yaklaşımdır.

Lindeberg'in (Lindeberg, 1994) çalışması da, doğru bir ölçekten bağımsızlık için σ^2faktörü ile Laplace normalizasyonu yapılması gerekli olduğunu göstermiştir.

Mikolajczyk (Mikolajczyk, 2002) tarafından gösterildiği üzere ise, ölçek-normalize Laplace-Gaussianile hesaplanan en büyük ve en küçük uçdeğerler yöntemi, Hessian ve Harris algoritmaları gibi diğer yöntemlere göre daha kararlı öznitelikler üretilmesini sağlamaktadır.

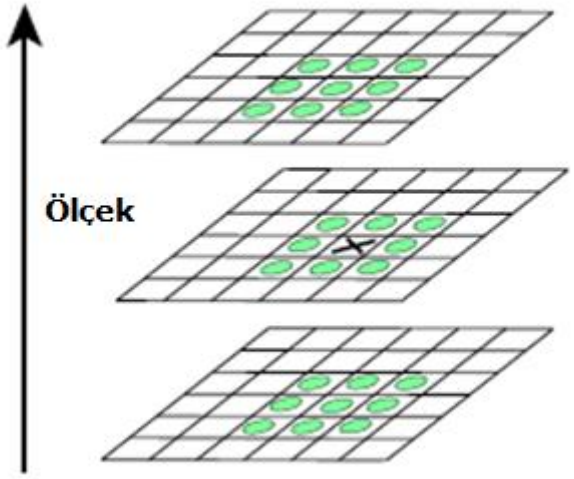
Şekil 2.2 de D(x,y,σ), DoG fonksiyonunun nasıl oluşturulduğu gösterilmektedir. Soldaki görüntünün ölçek uzayındaki her bir oktavı, Gauss ile tekrar tekrar konvüle edilir. Sağdaki DoG görüntülerini elde etmek için komşu Gauss görüntüleri çıkarılmıştır.

SIFT yaklaşımının en etkin özelliklerinden biri de herhangi bir görüntüdeki gri tonlu yerel örüntü özniteliklerini farklı bir ölçek uzay ayrım yordamı ile çıkarabilme yeteneğine sahip olmasıdır.



Şekil 2.2 : Soldaki Gauss ile konvüle etme, sağda Gauss çıkarılmış DoG görüntüleri.

Uçdeğerlerin konumlarını belirlemek için pikseller komşuları ile (kendi ölçeğinde 8 bir üst ve bir alt ölçekte 9 ar olmak üzere 26 komşu) karşılaştırılır ve en büyük ya da en küçük değer olan pikseller seçilir (Şekil 2.3).



Şekil 2.3 : Uçdeğer belirlemede kullanılan komşuluk ilişkisi

2.1.2 Anahtar nokta konumlandırma

Ölçek uçdeğer kestirimi ile çok sayıda anahtar nokta adayı tespit edilir. Fakat bu adayların büyük çoğunluğu kararsız yapıdadır. Bunun yanı sıra anahtar adaylarının bazıları tam olarak gerçek piksellerin üzerinde bulunmazlar ve çoğunlukla piksellerin arasında konumlanırlar. Bu sebeple, adayların ölçek uzayındaki gerçek konumlarının tespit edilmesi için bulundukları konumlara en yakın piksellere bakılarak ara değer kestirimi (interpolation) yapılır.

Bu aşamada her bir anahtar nokta adayının gerçek konumlarını tespit etmek için uçdeğerlerin ara değer hesabıyla belirlenen adreslerine göre yeniden konumlandırma yapılır (Lowe, 2004). Lowe'nin çalışmaları bu yaklaşımın eşleme kabileyetinde ve kararlılıkta artış sağladığını gözler önüne sermiştir. Söz konusu ara değer hesabı ise Difference-of-Gaussian (DoG) ölçek uzayı fonksiyonunun, D(x,y,σ), Taylor açılımı ile elde edilmiş hali kullanılarak yapılır:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \tag{2.4}$$

(2.4) eşitliğinde D ölçek uzayı fonksiyonunu, x bir yerel öznitelik noktasını göstermektedir. Bu eşitliğin x'e göre türevi alınarak aranan $\hat{x}$ uç ara değer fonksiyonu (2.5) eşitliği kullanılarak kestirilir ve ve anahtar adaylarının gerçek konumları bulunur.

$$\hat{x} = -\frac{\partial D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \tag{2.5}$$

Anahtar noktaların düşük kontrastlı olması muhtemel kararsızlığa yol açmaktadır. Kararsızlık, özellikle gürültüye karşı dayanıklılığı düşürdüğü için düşük kontrastlı aday noktaların elenmesi gerekmektedir. Düşük kontrastlı noktaların tespit edilmesi için 2.6'deki denklemden hesaplanan değerler 0.03 eşik değeri ile karşılaştırılır ve düşük kontrastlı adaylar elenir. Eğer çıkan sonuçlar 0.03'den küçük ise düşük kontrastlı sayılarak elenir.

$$D(\dot{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \tag{2.6}$$

Anahtar nokta adayları küçük gürültü değişimlerine karşı gürbüz olmasa bile, DoG fonksiyonu obje kenarları üzerinde çok güçlü cevap verir. Özellikle aynı kenar üzerinde belirlenen her noktanın birbirine benzemesi bu noktaların ayırt ediciliğini azaltarak karasızlığa yol açmaktadır. Bu sebeple kenarlar üzerinde yer alan anahtar

adaylarının elenmesi gerekmektedir.

DoG fonksiyonunda, kenar üzerinde bulunan noktalar yüksek temel eğimlilik (principal curvatures) geçişine sahiptir fakat dikey yönde bu değer küçüktür. Yüksek temel eğimlilik değerlerinin hesaplanması için 2x2'lik Hessian Matrisinden yararlanılır.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.7)$$

Eğimlilik değerleri uç değerin hesaplanması sonucu uçdeğerin konumu hakkında aşağıdaki bilgiyi verir:

- İki eğim de küçük ise uçdeğer düz bir konumdadır.
- Bir eğim büyük (kenara dik), diğer eğim küçük (kenar boyunca) ise kenar üzerindedir.
- İki eğim de büyük ise köşe üzerindedir

Bu işlemin sonucunda obje kenarları üzerindeki adaylar da elendiğinde yalnızca kararlı adaylardan oluşan anahtar noktalar elde edilir.

2.1.3 Anahtar noktalara bağımsız yönelimler atanması

Eşlemede kullanılmak üzere bulunan anahtar noktalara yönelim ataması yapabilmek için, belirlenen anahtar noktalar etrafındaki eğimlerin yönü ve genlikleri kullanılarak, en belirgin yönelim seçilir.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (2.8)$$

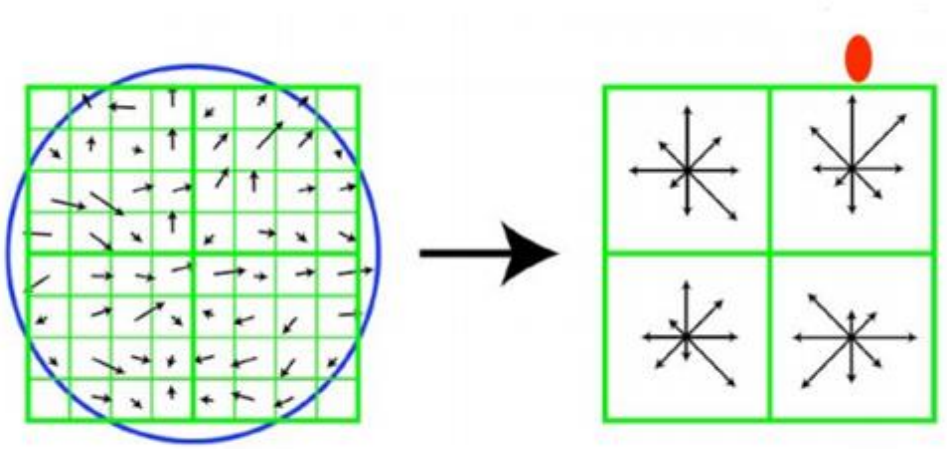
$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \quad (2.9)$$

(2.8) ve (2.9) formülasyonlarında sırasıyla (x,y) pikselindeki genlik ve yön formülasyonu verilmektedir. Eşitliklerdeki L(x+1,y), Gauss uygulanmış görüntünün (x+1, y) pikselinin gri düzeyini göstermektedir. (2.8) formülasyonunda görüldüğü gibi komşu piksellerin gri düzey farklılığı arttıkça genlik artmaktadır.

Anahtar nokta etrafındaki tüm piksellerin eğim yönelimleri kullanılarak, bir yönelim histogramı oluşturulur. Histogram ile en baskın yönelim veya yönelimler seçilir, gerekirse başka anahtar noktalar da belirlenir.

2.1.4 Anahtar nokta tanımlayıcıları (öznitelik vektörü) oluşturulması

Daha önceki aşamalarda görüntünün SIFT öznitelikleri ile belirlenen konum, ölçek ve yönelim bilgileri her bir anahtar noktaya yerleştirilmiş oldu. Bu bilgiler, görüntünün yerel bölgesini tanımlayan 2 boyutlu koordinat sistemini belirlemesi sebebiyle konum, ölçek ve yönelim parametrelerine göre değişmezlik (invariance) kazanılmasını sağladı (Lowe, 2004). Bu aşamada ise amaç elde edilen yerel görüntü bölgelerini doğrudan temsil eden, yüksek derecede ayırıcı yapıya sahip ve farklı koşullara göre gürbüz tanımlayıcılar oluşturmaktır.



Şekil 2.4 : Öznitelik vektörü oluşturma

Anahtar nokta tanımlayıcısı, Şekil 2.4'de solda gösterildiği gibi σ değerinin tanımlayıcının genişliğinin yarısına eşit olduğu yerlerde, Gaussian penceresi tarafından ağırlıklandırılan çember tarafından gösterilen gradyan büyüklüğünü ve uyumunu ölçerek yaratılır. Gaussian pencerelemenin amacı pencerenin pozisyonundaki küçük değişimlerle, tanımlayıcıdaki ani değişimleri önlemek ve tanımlayıcının merkezinden uzakta olan gradyanlara, yanlış giriş (misregistration) hatalarından en çok etkilenenler olduğu için, daha az ağırlık vermektir. 4x4 piksellik alt-bölgelerin üzerindeki içerikleri özetleyen yönelim histogramları (orientation histogram) Şekil 2.4'te sağ tarafta gösterildiği gibi bu örneklerden kurulur. Burada her bir okun uzunluğu bölge içerisindeki bu yöne yakın gradyan büyüklükleri toplamını simgeler. Bu resimde, 8x8 örnekleme setinden ölçülen 2x2 tanımlayıcı ok örnek olarak gösterilmiştir, ancak SIFT öznitelik hesabında 16x16 örnek okundan ölçülen 4x4 tanımlayıcılar kullanılır. Bu nedenle her bir anahtar nokta için 4x4x8=128 elementli öznitelik vektörü elde edilir.

Tanımlayıcı boyutunun 128 olması oldukça yüksek bir değerdir, ancak boyut arttıkça daha yüksek oranda ve daha düşük boyutluluğa sahip olarak görünebilir. Fakat yüksek-boyutlu tanımlayıcıların düşük-boyutlulara oranla daha yüksek oranda ve daha düşük hızda eşleme başarımı yakaladığı tespit edilmiştir. Tanımlayıcıların oluşturulması tek başına yeterli değildir. Işıklılık değişimi ve ölçek değişimi gibi durumlara karşı gürbüzlüğün sağlanabilmesi için öznitelik vektörleri üzerinde başka ek işlemler de uygulanmaktadır.

Işıklılık değişimlerinin etkisinin azaltılması için öznitelik vektörü birim uzunluğuna göre normalize edilir. Vektör normalizasyonu işlemi ile, her bir piksek değerinin bir sabitle çarpılmasıyla oluşan kontrast değişimleri engellenmiş olur. Bunun yanı sıra döndürmeden bağımsızlığı sağlamak için ise, anahtar noktanın açısı, her bir yönelimden çıkarılmalıdır. Böylece eğim yönelimleri, anahtar noktanın yönelimine göre, göreceli olacaktır.

2.2 Görüntülerin öznitelikleri kullanılarak eşleme yapılması

Farklı görüntülerden elde edilen anahtar noktaların birbirine uyumlu olup olmadığını denetlemek için kullanılan metotlardan birisi (2.10) ile tanımlı Öklid uzaklığıdır (Euclidean distance).

$$d(a,b) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \qquad (2.10)$$

Ancak bu hesabı yapmak yerine bundan daha az işlem yaparak, yakın sonuçlar elde edebilmek için Lowe farklı bir yöntem kullanmıştır. Bu yönteme göre bir görüntüden elde edilen öznitelik vektörü matrisinin her bir satırı, ikinci bir görüntüden elde edilenin transpozesiyle çarpılmıştır. Böylece (1*128) boyut ile (128*1280) boyutluk matris çarpıldığı için sonuçta (1*1280) boyutlu bir matris elde edilmiştir. Burada 1280, örnek için kullanılan görüntüdeki anahtar nokta sayısıdır. Elde edilen bu matrisin değerlerinin kosinüs tersi alınarak bulunan açılar sıralanır. En küçük açı, kendinden sonra gelen açının bir parametre ile çarpımından küçük ise bu anahtar nokta eşleştirilir. Aksi takdirde eşleştirme olmaz. Bu işlem sorgu SIFT öznitelik matrisinin tüm satırları kullanılana kadar devam eder. Bir başka deyişle sorgu görüntüsünde bulunan anahtar noktalar tek tek, diğer görüntüdeki anahtar noktalarda aranır. Tanımlanan eşleme yöntemi Öklid uzaklıpına göre kapsamlı arama yapmaktan daha kolay olmakla birlikte uygulamada çok yüksek işlem karmaşıklığına sahiptir. Bu nedenle literatürde birçok

ağaç tabanlı eşleme yöntemi önerilmiştir. Bu tez çalışmasında da ağaç tabanlı bir indeksleme ve eşleme yöntemi önerilmektedir. Önerilen yöntemi bir sonraki bölümde anlatılmaktadır.

3. İÇERİK TABANLI MOBIL GORUNTU ARAMA EŞLEME AĞACI

Görsel olarak birbirine yakın görüntülerin depolandığı büyük veri tabanlarında sorgu yapmak, son yıllardaki en zorlu araştırma konularından biridir. Cep telefonu ile çekilen bir sorgu görüntüsünü, milyonlarca orijinal referans görüntü içeren bir veri tabanında sorgulama gibi bir senaryo düşünebiliriz. Buradaki zorluk orijinal görüntülerin yanı sıra sorgu görüntülerinin ışık, yansıma, hareket vb. gibi sebeplerden dolayı referans görüntülerden farklılık göstermesidir. Özellikle klasik web aramalarından farklılaşan bu durum için yaygın olarak arama ağaçları şeklinde tasarlanmış görsel indeksleme yapıları kullanılmaktadır.

Bu bölümde görüntü arama ve eşleme için gerekli temel bileşenler olan görüntü indeksleme, arama-eşleme yöntemleri ile önerilen veri tabanı modeli ve veri tabanı arama ve eşleme metodu anlatılmaktadır.

3.1 Veri tabanı modelleri

Günümüzde teknolojinin ilerlemesi ve iletişim altyapılarının hızlanması ile veri akışı çoğalmakta ve bu veriyi yönetme ihtiyacı her geçen gün artmaktadır. Bu veriler çok faklı tipte ve amaçta üretilmektedir. Verinin yapısına ve kullanım amacına bağlı olarak bu verileri organize bir şekilde yönetmek amacıyla birçok veri yönetme yöntemi önerilmiştir. Bazı veriler yazılı, bazı veriler görsel olabilir ya da veri tipinden bağımsız büyük ya da az olabilir. Veri tabanı kavramı bu ihtiyaçtan doğmuştur ve verinin organize bir şekilde tutulması anlamına gelmektedir. Bu verileri tutmak için birçok veri yapısı ve veri tabanı mimarileri bulunmaktadır (Elmasri, 2011).

Veri tabanlarını 4 ana kategoride değerlendirebiliriz (Elmasri, 2011).

- Yazı ve sayısal verileri tutan geleneksel veri tabanları
- Ses, video, görüntü gibi verilerin tutulduğu multimedya veri tabanları
- Harita vb. bilgilerin tutulduğu coğrafi bilgi sistemi veri tabanları
- Son 10 yılda gündeme gelen, dünya üzerinde farklı lokasyonlarda dağıtık bir

biçimde veriyi tutmayı gerektiren büyük veri (big data) veri tabanları ve mimarileri

Geleneksel veri tabanları verilerin satır ve sütunlardan oluşan tablolarda tutulduğu ve yönetildiği veri tabanları yapılarıdır. Bu tablolar arasındaki veriler birbirleri ile ilişkili olabilmekte ve sorgulama işlemi sırasında indeksleme ihtiyacı oluşabilmektedir. Bu gibi ihtiyaçlardan dolayı, ilişkisel veri tabanı yönetim sistemi RDBMS (Relational Database Managemen System) adı verilen veri tabanını yönetmeye yarayan sistemler/yazılımlar bulunmaktadır.

Özellikle bu tez çalışması kapsamında görüntü içeriklerinin olması ve büyük veriler arasında arama/eşleme problemi üzerinde durulması sebebi ile multimedya ve büyük veri veri tabanı kategorileri çalışma ile ilgilidir.

Multimedya veri tabanları multimedya veri tiplerindeki verilerin tutulması ve organize edilmesi ile ilgilidir (Yu, 2011). Bu veri tipleri daha çok yazı, görüntü, grafik nesne, ses, video, animasyon tipindeki veri yapılarıdır. Bu verileri destekleyen veri tabanı yazılımları mevcuttur. Multimedya veri tabanlarında en büyük zorluk devamlı güncellenen multimedya veri tipleridir. Bu sebeple devamlı yazılımları güncelleme ihtiyacı oluşmaktadır.

Büyük veri (big data) kavramı, büyük ve karmaşık veri setlerinin geleneksel veri işleme süreçlerinin yetmediği durumlarda ortaya çıkmış bir kavramdır. Son dönemde mobil cihazların gelişmesi ile birlikte üretilen verideki artış, bu veriyi saklama, işleme, sorgulama ve sıkıştırma problemlerini gündeme getirmiştir. Özellikle her geçen gün internetin yaygınlaşması, arama motorlarındaki indekslenen verilerin büyümesine ve zorlaşmasına sebep olmaktadır. Bulut teknolojilerinin de gelişmesi ile her hangi bir lokasyondaki veriyi her hangi bir lokasyonda işleme kolaylaşmıştır.

Büyük verilerdeki en büyük zorluklardan biri de verilerin yapılandırılmamış (unstructured) olmasıdır. Yapılandırılmış veriler daha çok klasik satır sütün verileri gibi düzenli ve boyutu belli verilerdir. Yapılandırılmamış veriler ise belirli bir patern doğrultusunda oluşturulmamış, birçok karmaşık veriyi arasında barındıran verilerdir. Yapılandırılmamış verilere en iyi örneklerden biri mesajlaşma verileridir. Mesajlaşma uygulamalarında üreyen mesaj verileri kullanıcılar tarafından belirli bir patern doğrultusunda üretilmemektedir. Bu veriler kullanılarak örneğin iletilen bilgide en çok hangi içeceğin adı geçtiği bilgisi aranması için tüm yazışmaları alıp işlemek ve özet

bir bilgi çıkarmak gereklidir. Milyarlarca yazışma olduğunu düşünürsek işlenmemiş bu verilerin dağıtık ve verimli bir süreç ile işlenmesi gerekmektedir. Bu verilere hızlı erişim için verilerin işlenerek bir özet veri (metadata) tutulması ve verilerin indekslenmesi gerekmektedir.

Büyük veri işleme çalışmalarında tüm veriye gitmek zaman alıcı ve maliyetli olacağı için çeşitli veri modelleri ve indeksleme yöntemleri üzerinde durulmaktadır. Bu yöntemlerden öne çıkanlardan biri hiyerarşik veri modelleridir. Hiyerarşik veri modelleri ağaç benzeri bir yapıda modellendiği veri modelleridir. Bu tez çalışmasında kullandığımız sözcük ağacı da bu veri modeline dayanmaktadır.

3.2 Sözcük Ağacı Oluşturma ve Indeksleme

Referans görüntülere ilişkin özniteliklerin tamamının veri tabanında saklanması ve sorgu görüntüsüne ilişkin özniteliklerin veri tabanında kayıtlı bulunan bütün referans özniteliklerle karşılaştırılması şeklinde gerçeklenen kapsamlı arama (exhaustive search), işlemsel yükün çok fazla olması nedeniyle, kullanışlı değildir. Bu amaçla geliştirilen çözümlerin en çok kabul görenlerinden birisi özniteliklerin saklanmasında sözcük ağaçları kullanılmasıdır (Nister, 2006). Sözcük ağaçları temel olarak içerisinde bulunan sözcükleri birbirlerine benzerliğine göre hiyerarşik bir biçimde gruplayıp saklama temeli üzerine kurulur.

Bu çalışmada, görüntüler sözcüklere karşı düşürülmekte ve her bir sözcük ilgili SIFT öznitelik vektörleri ile temsil edilmektedir. Görüntülerin sözcük ağacına yerleştirilmesindeki temel fikir, benzer görüntülerden/görüntü parçacıklarından benzer özniteliklerin çıkacak olması ve benzer özniteliklerin sözcük ağacı üzerinde yakın yerlere konumlandırılmasıdır. Bu amaçla veri tabanında saklanacak görüntülerden çıkarılan SIFT özniteliklerinden öznitelik gruplama BoF (bag-of-features) yöntemi ile sözcüklere karşı düşen öznitelik kümeleri bulunmaktadır. Bu çalışmada öznitelik gruplama hiyerarşik k-ortalamalı kümeleme (k-means clustring) ile gerçeklenmiştir.

Sözcük ağacı kullanarak arama ve eşleme yapmak için temel olan 3 ana adım gerçekleşirtirilir.

- Referans görüntülerin özniteliklerinin çıkarılarak hiyerarşik k-ortalamalı öbekleme algoritması ile öbek ortalamalarıyla belirlenen sözcüklerin elde edilmesi ve veri tabanında indekslenmesi

- Sorgu görüntüsü özniteliklerinin en yakın sözcük ilişkisi izlenerek ağaca yerleştirilmesi
- Skorlama yapılarak referans görüntüler arasından en iyi eşlenen görüntünün getirilmesi.

3.2.1 Hiyerarşik k-ortalamalı kümeleme ile sözcük ağacı oluşturulması

K-ortalamalı öbekleme algoritması kümeleme probleminde kullanılan yaygın algoritmalardan biridir. Algoritma, öznitelikleri çıkarılmış bir veri seti üzerinde k adet kümeleme gerçekleştirir. Arama ve eşleme çalışmalarında kullanılan sözcük ağaçlarında, görüntüler sözcüklere karşı düşürülmekte ve her bir sözcük ilgili SIFT öznitelik vektörleri ile temsil edilmektedir. Algoritmanın arka planında yatan temel fikir k sayıdaki her bir küme (grup) için bir merkez tanımlanması ve bu merkezlerin mümkün olduğunca birbirinden uzak olacak şekilde yerleştirilmesi, konumlandırılmasıdır. Amaç küme içi benzerliğin maksimum, kümeler arası benzerliğin ise minimum tutulmasıdır.

Bunun için veri setinden her bir kümeyi temsil edecek merkezleri (centroid) temsil etmek üzere k adet öznitelik rastgele seçilir. Sonraki aşamada ise veri setindeki geri kalan öznitelikler, daha önce belirlenmiş merkezlere olan uzaklıkları karşılaştırılarak kendilerine en yakın kümelerle ilişkilendirilir. Bu işlem bittikten sonra kümelerdeki her bir noktanın küme ortalamasına uzaklıkları hesaplanarak o anda en yakın olduğu öbeğe atanır ve yeni elemanlarla öbek merkezleri yeniden hesaplanır. Bu işlem merkezler yer değiştirmeyene kadar sürdürülür.

$X = \{x_1, x_2, x_3, ..., x_n\}$ öznitelikleri ve $V = \{v_1, v_2, ..., v_c\}$ öbek ortalamalarını temsil edecek şekilde k-ortalamalı öbekleme algoritmasında denklem 3.1 de görülen maliyet fonksiyonu minimize edilir.

$$J(v) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\left| \left| x_j - v_i \right| \right| \right)^2 \tag{3.1}$$

- $||x_j - v_i||$, x_j ve v_i öznitelik vektörleri arasındaki Öklid uzaklığı,
- *'c_i', i.* kümedeki öznitelik sayısı,
- *'c',* toplam merkez (centroid) sayısı

Kümelerdeki özniteliklerin ortalaması alınarak elde edilen merkezlerdeki güncelleme

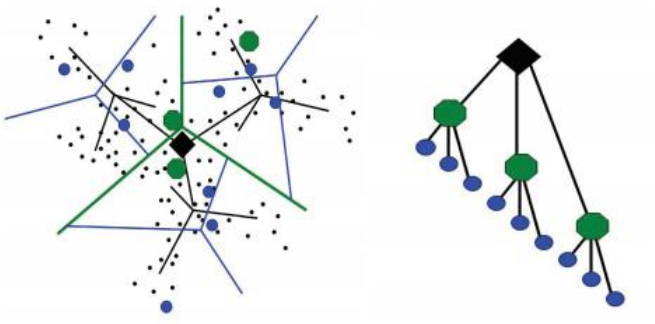
işlemi ise, aşağıdaki formül kullanılarak güncellenen öbek merkezleri denklem 3.2 ile hesaplanır.

$$v_i = \left(\frac{1}{c_i}\right)\sum_{j=1}^{c_i} x_i \tag{3.2}$$

Yukarıda anlatılan k-ortalamalı öbekleme algoritması kullanılarak gerçekleştirilen öbekleme işlemi ağaç yapısı oluşturmak için yeterli değildir. Bu sebeple hiyerarşik kümeleme işlemi uygulanmıştır.

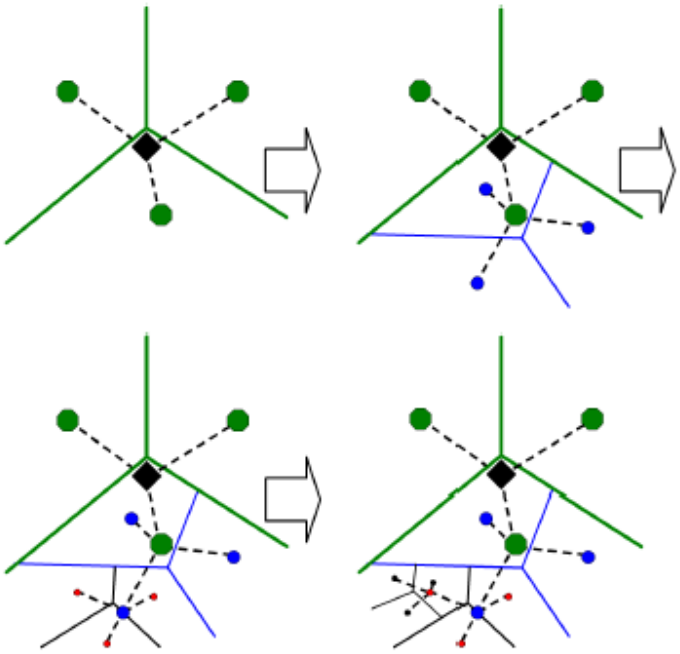
Hiyerarşik kümeleme yapılırken kullanılan iki önemli parametre tüm ağaç yapılarında olduğu gibi ağacın dallanma katsayısı (branching factor) 'k' ve ağacın derinliği (depth) 'l' olarak gösterilebilir.

Hiyerarşik kümeleme, hiyerarşik olmayan k-ortalamalı kümelemeye oldukça benzemektedir. Farklı olarak bu kez k sayısı dal sayısını ifade ederken, her bir düğümde ise öznitelik vektörlerinin uzaklık ortalamalarından elde edilen ve Denklem (3.2) de tanımlanan küme merkezleri (centroid) tutulur. Hiyerarşik k-ortalamalı kümeleme yapılırken her adımda 'k' ortalama sayısı dallanma katsayısı olarak alınarak ağacın tüm seviyelerinde anne düğüme ait çocuk düğüm sayısı eşit tutulur. Bu şekilde oluşturulan her bir kümede kendi içerisinde yine 'k' adet kümeye ayrılır. Bir önceki bölümde verilen örnekte öznitelikler iki boyutlu düzlemde dağınık olarak kümelenirken hiyerarşik kümelemede öznitelik vektörleri ağacın yapraklarına yerleşir. Şekil 3.1'de, k=3 olacak şekilde hiyerarşik öbekleme ile ağaç oluşturma örneği gösterilmiştir.



Şekil 3.1 : k=3, solda üstten görünüm, sağda ağaç görünümü (Nister, 2011).

Her bir küme kendi içinde yine 'k' adet kümeye ayrılırken her ayrım veya dallanma sözcük ağacı içerisinde yeni bir seviye (layer) oluşması anlamına gelmektedir. Tüm ağaç boyunca L maksimum seviyesine kadar seviyeler oluşur. Şekil 3.2'de sözcük ağacı oluşturma sürecinin bir görsel anlatımı bulunmaktadır. Hiyerarşik kümeleme ile k = 3 olacak şekilde her seviyenin oluşumu görülmektedir. Kök düğümden bir alt seviyeye dallanırken 3 çocuk dal oluşmakta, alt seviyede oluşan her bir çocuk dal için bir alt seviyede yeniden 3 çocuk dal oluşmaktadır.

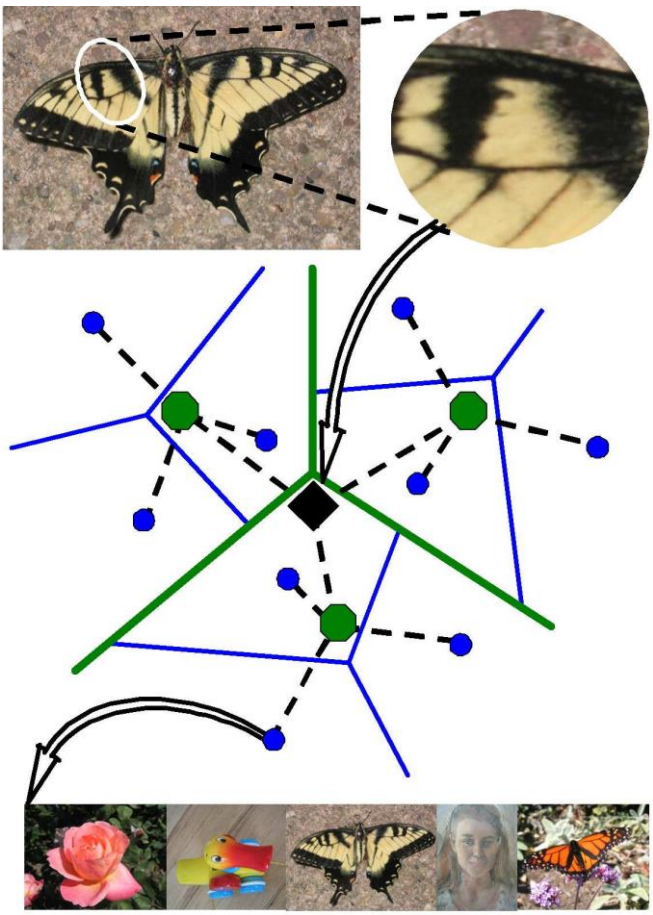


Şekil 3.2 : Sözcük ağacı oluşturma görsel anlatımı (Nister, 2006).

Her bir öznitelik vektörü, ağaca yerleştirilirken her seviyede bir alt seviyeye geçmek için 3 dal merkezi ile karşılaştırılarak en yakın dala dallanarak ilerlemektedir. Bu tez çalışması kapsamında k = 3 ve l = 3 olacak şekilde sözcük ağaçları oluşturulmuştur.

Bu amaçla kare ya da dikdörtgen çok sayıda bölge görüntüden çıkarılmaktadır. Her bir bölge için SIFT öznitelik vektörleri bulunmaktadır. Bu öznitelik vektörlerleri hiyerarşik olarak sözcük ağacı içerisinde öbeklenmektedir. Sözcük ağacındaki her bir düğüm ters indeks kullanılarak gerekli bilgiyi tutmaktadır. Her bir ağaç düğümünde ters indeks yöntemi ile o düğümden geçen referans görüntü indeks bilgisi (Image ID) ve o referans görüntünün kaç özniteliğinin düğümden geçtiği yani frekans bilgisi veri tabanında saklanmaktadır (Chen, 2010). Şekil 3.3 de yerel özniteliklerin ağaç düğümleri ile hiyerarşik eşlenmesi gösterilmeye çalışılmıştır. En alt seviyedeki yaprak düğümde birden çok referans görüntüsüne ait öznitelikler bulunabilmektedir. Şekil 3.3

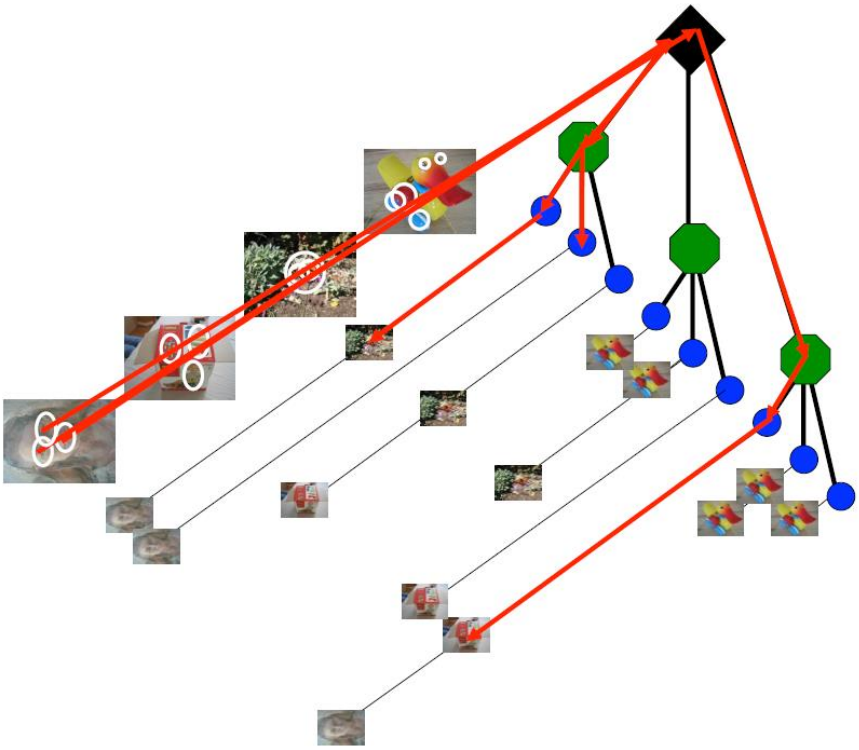
den de görüleceği gibi yaprak düğümlerinde 5 referans görüntüye ait öznitelikler bulunmaktadır.



Şekil 3.3 : Özniteliklerin ağaç düğümlerine yerleştirilmesi

3.2.2 Düğüm indeks bilgileri

Tüm referans görüntüler sözcük ağacına yerleştirildikten sonra, eşleme aranması için gelen sorgu görüntüsünün öznitelikleri çıkarılarak ağaca yerleştirilmesi gerekmektedir. Ağaca yerleştirme işlemi, sorgu görüntüsü özniteliklerinin kök düğümden başlayarak her bir düğümde kayıtlı olan sözcük bilgisine en yakın olan yani en düşük Öklid uzaklıklı olduğu düğüm üzerinden dallanarak devam eder. Şekil 3.4'te dört farklı sorgu görüntüsünden çıkarılan SIFT özniteliklerinin kök düğümden itibaren katmanlı olarak sözcük ağacına yerleştirilmesi gösterilmektedir. Ağacın alt seviyelerine inildikçe görüntüye ilişkin ayırt edicilik artmakta ve yaprak düğümlerde çoğunlukla tek bir sorgu görüntüsüne ait lokal görüntü parçasını modelleyen bir veya iki SIFT özniteliği kalmaktadır.



Şekil 3.4 : Ağaç düğümlerine yerleşmiş sorgu görüntü öznitelikleri

Tez çalışması kapsamında veri tabanı olaraka nesne tabanlı (object oriented) bir veri tabanı olan db4o kullanılmıştır. Java tabanlı olan db4o veri tabanında sözcük ağacı düğümü için cluster adını verdiğimiz nesne yapıları oluşturulmuştur. Cluster nesnesi içerisinde her bir düğümde aşağıdaki bilgiler kaydedilmektedir:

- Düğüme ait indeks id bilgisi
- Düğümden geçen referans görüntülerin indeks id bilgisi,
- Referans görüntünün kaç özniteliğinin bu düğümden geçtiği bilgisi,
- Düğümden geçen SIFT öznitelik vektörleri
- Düğüme ait entropi değeri
- Düğüme ait 128 bit öbek ortalaması vektörü ile modellenen sözcük bilgisi
- Düğümün ağaç içerisinde bulunduğu seviye (layer) ve o seviyedeki kaçıncı düğüm olduğunu belirten sıralama bilgisi,
- Bir üst seviyedeki anne düğüm bilgisi
- Bir alt seviyedeki çocuk düğüm id'leri bulunmaktadır.

Sözcük ağacı, iç içe dallanan ve farklı nesne tiplerini içeren bir yapıya sahiptir. Bu sebeple nesne tabanlı veri tabanları, yapısı gereği, sözcük ağacındaki sözcükleri birer nesne olarak temsil etmeye uygundur. Bir alterantif olarak XML tabanlı veri tabanları da tercih edilebilir. XML tabanlı veri tabanlarında tüm verinin XML yapsı içersinde işaretlenmiş (tag) olması ve nesnelerin iç içe yapılar şeklinde tutulabiliyor olması, sözcük ağacı uygulamalarında kullanımlarını avantajlı kılmaktadır. Fakat büyük verilerde XML dosyanın boyutunun büyümesinden dolayı sorgu performanlarında düşme olmaktadır ve sorgular zaman almaktadır. Nesne tabanlı veri tabanlarında veriler nesne şeklinde tutulmaktadır. Nesnelerin içerisindeki değişkenlerde (variable) saklanmak istenen veriler esnek bir şekilde güncellenebilmektedir. Örneğin, çalışma süresince sözcük ağacında yukarıda bahsedilen indeks bilgilerine ekleme yapılması ihtiyacı doğmuş ve cluster nesnesi içerisinde yaratılan değişkenler ile kolayca eklenmiştir. Kullanılan db4o nesne tabanlı veri tabanındaki indeksleme ve önbelleğe alma (caching) özelliği sayesinde sorgulama performansı arttırılmıştır.

Bu tez çalışması kapsamında kitap, CD, DVD ve sanatsal tabloların mobil cihazlar ile çekilmiş orijinal görüntüler kullanılarak dört adet veri tabanı ile performans testleri gerçekleştirilmiştir. Tez çalışması, aşağıdaki yazılım geliştirme ortamları kullanılarak gerçeklenmiştir.

- Geliştirme Ortamı: Eclipse SDK 3.7.0,
- Geliştirme Versiyonu: JDK: JAVA SE 1.6
- Veri tabanı Ortamı: DB4O (db4o-8.0.224.15975-core-java5.jar)
- Geliştirilen İşletim Sistemi: Ubuntu 11.04 & Mac OS X Yosemite 10.10.2

4. MOBİL GÖRÜNTÜ ARAMA VE EŞLEME ALGORİTMALARI

Sorgu görüntüsünün, çok geniş bir veri tabanı üzerindeki referans görüntülerle ikili öznitelik karşılaştırma (pairwise comparison) metodu kullanarak eşlenmeye çalışılması oldukça elverişsiz bir yöntemdir. Bunun yerine, veri tabanındaki eşleşmesi daha muhtemel referanslardan oluşan adayları tutan bir veri yapısının oluşturulması tercih edilir. Yerel öznitelikleri indekslemede kullanılabilecek çok sayıda veri yapısı mevcuttur. Bu çalışmada ise referans görüntülerden elde edilen öznitelikler üzerinde hiyerarşik k-ortalamalı kümeleme kullanılarak oluşturulmuş sözlük ağacı yapısı kullanılmıştır. Bunun en önemli sebeplerinden biri ise, sözcük ağacı yapısı üzerinde arama-getirme (search-retrieve) işlemini hızlandıracak ağırlıklandırma metodunun kolaylıkla uygulanabilmesidir. Öncelikle ağaçtaki her bir düğüm i için entropi değerleri kullanılarak ağırlık vektörü w_i tanımlanır. Denklem (4.1) sorgu görüntüsüq_i ve Denklem (4.2) veri tabanındaki referans görüntü d_i için ağırlıklandırma işlemini ifade etmektedir.

$$q_i = n_i w_i \tag{4.1}$$

$$d_i = m_i w_i \tag{4.2}$$

n_i : sorgu görüntüsü için öznitelik vektörü sayısı

m_i : referans görüntüsü için öznitelik vektörü sayısı

Veri tabanındaki toplam görüntü sayısı N ve bir düğümden geçen birbirinden farklı görüntü sayısı N_i ise, w vektörünün i. elemanı (4.3)'deki gibi hesaplanır.

$$w_i = \ln \frac{N}{N_i}, \quad i = 1, \dots, N_c \tag{4.3}$$

Sözlük ağacı yukarıda anlatılan işlemlerle şekilde oluşturulduktan sonra nesne tabanlı bir veri tabanında saklanır. Nesne yapısının, ağaç yapısındaki verileri doğrudan temsil

etmeye uygun olması ve üzerinde gerçekleştirilen sorgu işlemlerinin çoğul ortama uygun olması sebebiyle veri tabanının bu yapı ile oluşturulmasına karar verilmiştir.

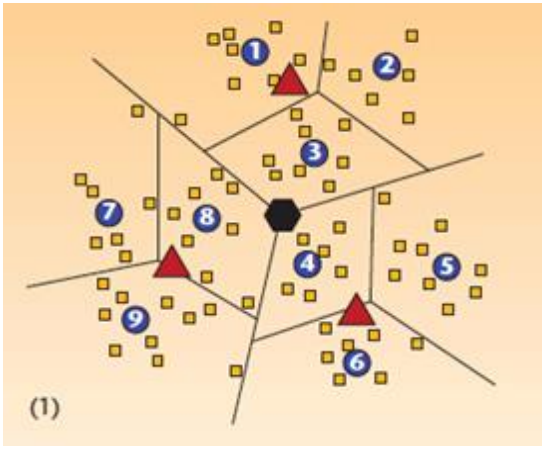
4.1 Görsel Benzerlik Skorlama

Veri tabanındaki referans görüntülerin vektörleri ağırlıklandırma yapılarak oluşturulmuş ve daha önce veri tabanında saklanmıştı (4.2). Sorgulama sırasında gelen sorgu görüntüsüne ait vektör, referans görüntülerde uygulanan yöntemle aynı şekilde önce ağaç oluşturularak daha sonra da ağaç üzerinde ağırlıklandırma yapılarak elde edilir (4.1). Bu adımdan sonra ağaç veri tabanı üzerinde arama işlemi başlar. Sorgu görüntüsü referans görüntülerden kendisine en fazla benzeyenle eşleşme eğilimi gösterecektir. Kendisine en fazla benzeyen referansa karar verebilmesi için ise arama-karşılaştırma esnasında skorlama yapılır. Sorgu görüntüsü ve referans görüntü arasındaki skor s ise daha önce elde edilen sorgu ve referans vektörlerinin birbirlerine olan uzaklıklarının hesaplanması ile elde edilir (4.4).

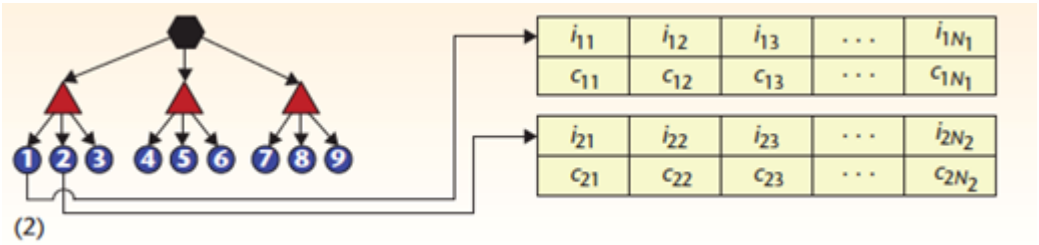
$$s(q,d) = \left|\left|\frac{q}{||q||} - \frac{d}{||d||}\right|\right| \tag{4.4}$$

4.2 Ters İndeks (Inverted Index)

Ters indeks (inverted index) veri yapısı özellikle arama motorlarında sıkça kullanılan bir tekniktir. Büyük veri tabanlarında hızlı görüntü eşleme gerçekleştirmek için, sözcük ağacı (VT) veri tabanında bulunan her ağaç düğümünü ziyaret eden görüntüleri ve her düğümdeki görüntü sayılarını eşleyen bir ters indeks dizini tutulmaktadır.



Şekil 4.1 : Hiyerarşik k-means kümeleme (Girod, 2011),



Şekil 4.2 : Ters indeks dizisi (Girod, 2011),

Şekil 4.1 de hiyerarşik k-means kümeleme görülmektedir. Sözcük ağacının L seviyesi olsun. Bu L seviyedeki kök düğüm dışında en alttaki veya en içteki düğümlerde C adet yaprak düğümü içersin. Bu şekilde bir sözcük ağacında $K = C^L$ adet yaprak düğümü bulunmaktadır. Örneğin L = 2, C= 3 olan bir ağaçta K = 9 adet yaprak düğümü bulunmaktadır.

Sözcük ağacı ile ilişkili ters indeksleme, Şekil 4.2 deki gibi her bir yaprak düğümü için iki tip dizi içermektedir. Her k düğümü için düğümü ziyaret eden Nk adet görüntü id bilgilerini içeren, $\{i_{k1}, i_{k2}, \dots, i_{kN_k}\}$ şeklinde sıralanmış bir dizi mevcuttur. Bu görüntü id'lerini içeren dizi ters indeks olarak adlandırılır.

Benzer şekilde her bir k düğüm için düğümü ziyaret eden Nk adet görüntünün ziyaretinin frekansını yani sayısını tutan bir eş boyutlu sayım dizisi $\{c_{k1}, c_{k2}, \dots, c_{kN_k}\}$ de bulunmaktadır.

Görüntülere ilişkin son kullanıcıları ilgilendirmeyen bir bilgi de ayrıca veri tabanında tutulur, bu da her bir görüntüye ilişkin 'dolaşı vektörü' (traverse vector) diye ifade edilen yapıdır. Her bir görüntüye ilişkin öznitelik vektörlerinin ağacın hangi düğümleri altında toplanmış olduğunu belirtir.

N adet görüntü içeren bir veri tabanında, sorgu sırasında, sorgu görüntü özniteliklerinin hangi düğümden geçtiği bilgisi dolaşı vektörü kullanılarak skorlanabilir.

s(i), i. veri tabanı görüntüsü için skorlama bilgisi olsun. Başlangıçta, herhangi bir düğümü ziyaret edilmediğinden s(i) değeri 0 olarak belirlenir. Sorgu görüntü öznitelikleri tarafından k. düğümün qk kez ziyaret edildiğini düşünelim. k düğümü için

ters indeksteki $\{i_{k1}, i_{k2}, \ldots, i_{kN_k}\}$ dizisi kullanılarak tüm görüntüler için skorlama denklem 4.5 kullanılarak hesaplanır.

$$s(i_{kj}) \coloneqq s(i_{kj}) + \frac{w_k^2 c_{kj} q_k}{\Omega_{i_{kj}} \partial_{i_{kj}}}, j = 1, \ldots, N_k \tag{4.5}$$

Denklem 4.5 de görülen w_k, bahsedilen k düğümü için ağırlıklandırma vektörüdür. $\Omega_{i_{kj}}$, i_{kj} id'sine sahip veri tabanı görüntüsü için normalizasyon faktörüdür (4.6), ve denklem 4.6 da görüldüğü gibi n düğümünden geçen veri tabanı görüntülerinin sayısıdır. $\partial_{i_{kj}}$, sorgu görüntüsü için normalizasyon vektörüdür (4.7) ve denklem 4.7 de görüldüğü gibi n düğümündeki sorgu görüntüsünün sayısıdır.

$$\Omega_{i_{kj}} = \sum_{n=1}^{K} W_n \tag{4.6}$$

$$\partial_{i_{kj}} = \sum_{n=1}^{K} W_n \tag{4.7}$$

Sorgu görüntüsü tarafından ziyaret edilen diğer düğümlerdeki skor hesaplaması da benzer şekilde yapılmaktadır. En yüksek s(i) skorunu alan veri tabanı görüntüleri en iyi eşleşen görüntüler olarak adlandırılır.

Ters indeksleme ile yapılan çalışmanın sonuçlarını 5. Bölümde bulabilirsiniz. (4.1) deki ağırlandırmaya ek olarak ters indeksleme ile görüntülerin her bir düğümdeki sayısı ve ziyaret sıklığı bilgisi skorlamaya eklenmiştir ve eşleme başarımında iyileştirme kaydedilmiştir.

Bu yöntem, skorlamaya ek bilgi getirdiği için daha fazla hafıza alanı ve sorgu sürelerinde artış anlamına gelmektedir.

4.3 Bağlamsal Ağırlıklandırma (Contextual Weighting)

Özniteliklerin ayrıştırıcı gücü, özdeş olanlar dahil görüntüden görüntüye değişmektedir. Halı, çim ve toprak gibi dokusal alanları fazla olan görüntülerin öznitelikleri, sayısal olarak fazla olmalarına rağmen az bilgi içermektedir. TF-IDF (Term Frequency – Inverse Document Frequency) ağırlıklandırmada, geniş görüntü

veri tabanlarında hesaplandıklarında oldukça fazla olan özniteliklerin IDF'leri, TF'lerine göre eşleme skorlarında daha fazla katkı sağlamaktadırlar. Bu fazla sayıdaki öznitelik sayımı başarımı düşüren sonuçları almaya neden olmaktadır. Başarımı düşüren bu gibi özniteliklerin ağırlıklarını düşürerek elimine etmek gerekir. Bu öznitelikler, yaprak düğümlerine pek düşmezler ve büyük veri tabanında hesaplamalara dahil edilmesi efektif bir yöntem değildir. (Wang, 2011).

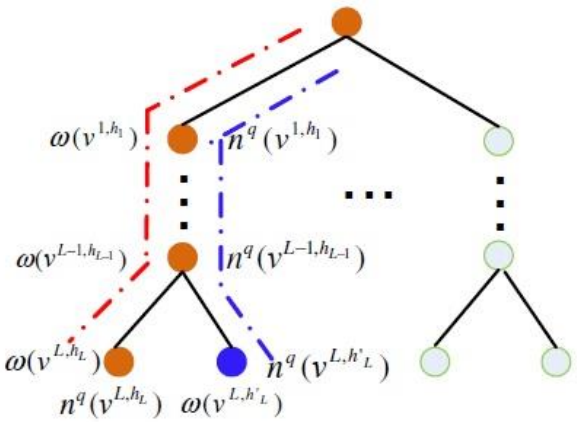
Bu gözlemden cesaretle, IDF'e ek olarak düğüme ait o düğümün altındaki tüm düğümlerin ters ağırlıklandırmalarının sayımlarını içeren izlenen düğüm yolu (node path) bilgisinin eklenmesi 'Bağlamsal Ağırlıklandırma' (Contextual Weighting) olarak adlandırılmıştır. v düğümündeki q sorgu görüntüsünden x_i.öznitelik ve veri tabanındaki d referans görüntüden y_j.öznitelik şeklinde varsayarsak $w(v)$ağırlıklandırması (4.8) denklemindeki şekildedir.

$$w_{i,j}^{q,d}(v) = w_i^q w_j^d idf(v) \tag{4.8}$$

q sorgu görüntüsünün v düğümündeki öznitelik sayısını n^q(v) olarak belirtelim. Sorgu görüntüsünün xi. özniteliğinin ağaçta yol boyunca izlediği düğüm sayısı ile oluşan w_i^q (4.9)'deki gibi hesaplanır.

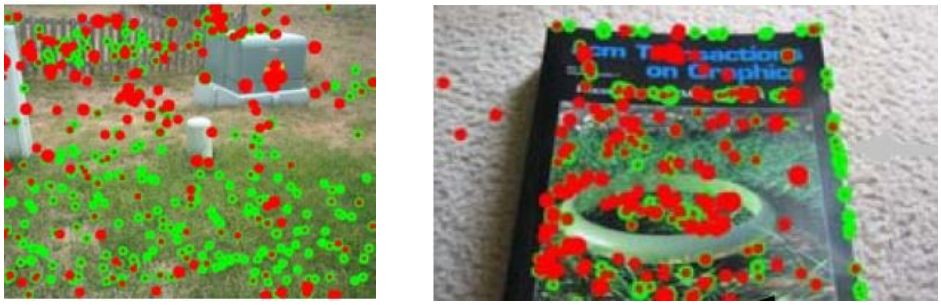
$$w_i^q = \sqrt{\frac{\sum_{v \in T(x_1)} \omega(v)}{\sum_{v \in T(x_1)} \omega(v) \times n^q(v)}} \tag{4.9}$$

$w(v), IDF(v)$ ile set edilen katsayıdır. Görüleceği gibi w_i^q ağırlığı sadece özniteliklere bağlıdır ve tüm T(xi) yolu boyunca dolaşılan v düğümleri için paylaşılmaktadır.



Şekil 4.3 : Öznitelik bağlamsal ağırlıklandırma örneği (Wang, 2011)

Şekil 4.3 den de görüleceği gibi, eğer iki öznitelik $x_i$ve x_i'sadece yaprak düğümde farklılaşıyorsa, (4.8)'deki denkleme göre özniteliklerin ortak ana düğümünün farklı ağırlıkları olacaktır. Sözcük ağacının karakteristiğine bağlı olarak düğüm yolunun ters ağırlıklandırma sayımı seçimi ile farklılaşacaktır. Örneğin, ağaç derin ise yaprak düğümlerinin çoğunun sadece bir öznitelik vektörü bir görüntü için quantize edilmiş olacaktır. (4.9) denklemine bu bakış açısı ile bakarsak komşu düğümlerde az sayıda olan bir öznitelik, eşlemede belirleyici olabilmektedir. Pratikte öznitelik bağlamsal ağırlıklandırmalar daha derin seviyelerdeki ağaç dallarında tespit edilmektedir. Şekil 4.4 de yeşil noktalar orijinal IDF ağırlıklarını, kırmızı noktalar öznitelik bağlamsal ağırlıklandırmaları göstermektedir. Şekil 4.4 de öznitelik bağlamsal ağırlıklandırmanın etkileri görülmektedir. Küçük kırmızı noktalar hafif ağırlıklandırılmış, yeşil noktalar ise ağır ağırlıklandırılmış özniteliklerdir.



Şekil 4.4 : Öznitelik bağlamsal ağırlıklandırmanın etkisi (Wang, 2011).

Arama yapabilmek için oluşturulan görüntü veri tabanlarındaki görüntülerin her bir ilgi noktası için öznitelikleri çıkarılarak 128 boyutlu vektör halinde veri tabanında saklanmalıdır.

4.4 Önerilen Arama ve Eşleme Yöntemi

Eşleme başarımını arttırmak için ağaçta izlenen yol (path) bilgisine istinaden, yeni bir yaklaşım olarak sorgu özniteliklerinin gittikleri en son yaprak düğümlerini baz alarak, ağacın katmanları arasında tüm ağaca yayılmış dağıtık düğümler de eşlemede ayırdedici etken olarak kullanılabilir. Tez kapsamında önerilen yöntem ile eşleme başarımı arttırılırken arama ve getirme karmaşıklığı da sadece yaprak düğümlerde işlem yapılması sebebiyle azaltılmaktadır.

Sorgu görüntüsünün ağaca yerleştirilmesi sonrasında, özniteliklerinin yerleştikleri en son yaprak düğümleri hesaba katalım. Bu düğümlerden sorgu görüntüsünün kaç adet

öznitelik vektörünün geçtiği bilgisi ile beraber bir sıralama dahilinde sorgu ile herbir veri tabanı görüntüsü arasında bir eşleme skoru S_d^q hesapladığımızı düşünelim.

Skorlama yapılırken her v_k düğümünden geçen *d* referans görüntüsü için skorlamaya katkı bileşeni$S_{q,d}(v_k)$ hesaplanarak *d* görüntüsüne ait skorlama bilgileri toplanmakta ve *d* görüntüsü için bir toplam eşleme skor değeri $S_d^q(v_k$) denklem (4.10)'deki gibi hesaplanmaktadır.

$$S_d^q(v_k) = \frac{1}{\sqrt{E}} \sum_{k=0}^{t-1} S_{q,d}(V_k) \tag{4.10}$$

(4.11) formürülünde görülen *t*, *q* sorgu görüntüsü ve *d* veri tabanı görüntüsüne ait özniteliklerinyerleştiği ortak yaprak düğüm sayısını ifade etmektedir.Formüldeki*E* değeri, skorlama hesaplamalarında değerlerin indeks ağacının boyutuna bağlı olarak ölçeklenebilir olması açısından ayırt ediciliği arttırmak amacıyla kullanılmaktadır. *E* denklem (4.11)'deki gibi hesaplanır.

$$E = \sum_{i=0}^{c} \omega(v_i) \tag{4.11}$$

(4.11) formülündeki *c*, veri tabanındaki toplam düğüm sayısını, $\omega(v_i)$ değeri, *i.* düğümün entropisini göstermektedir.

Denklem (4.10)'de görülen ve ortak herbir yaprak düğümün skorlamaya katkısını belirleyen $S_{q,d}(V_k)$ denklem (3.14)'te tanımlanmaktadır.

$$S_{q,d}(v_k) = w_k^q w_k^d idf(v_k) \tag{4.12}$$

(4.12) formülündeki $idf(v_k)$, v_k düğümündeki entropi değerini, $w_k^q q$ sorgu görüntüsü için v_k düğümünde hesaplanan ağırlık fonksiyonunu, $w_k^d d$ referans görüntüsü için v_k düğümünde hesaplanan ağırlık fonksiyonunu ifade eder. Sorgu ve veri tabanı referans görüntüsüne ait ortak yaprak düğümlerin indeksi ise *k* ile ifade edilir.

$$w_k^q = \sqrt{\frac{\sum_{i=0}^{z} \omega(v_i)}{\sum_{i=0}^{k} n^q(v_i)}} \tag{4.13}$$

w_k^q (4.13) denklemi ile hesaplanır. Bu formülasyonda payda görülen değer, *q* sorgu görüntüsünün gittiği tüm *z* yaprak düğümlerinin düğüm ağırlık değerlerinin (entropy) toplamıdır. Entropi toplamının sorgu görüntüsü için bir kez hesaplanması yeterlidir.

(4.13) formülasyonunda sorgu görüntüsünün yerleştiği her *i.* yaprak düğümdeki öznitelik sayısı veya frekansı $n^q(v_i)$ şeklinde ifade edilir. Önerilen formülasyonda farklı veri tabanı görüntüleri için eşlemede ayırd ediciliği sağlayan değişken kısım paydadaki toplamsal terimdir. Bu terim,sorgu görüntüsünün yaprak düğümden geçen öznitelik sayılarının işleme *i*=0 olarak döngüye sokulan ilk ortak yaprak düğümden başlayarak işleme sokulan*k.*ortak yaprak düğüme kadar toplanması ile elde edilir.Benzer şekilde sorgu görüntüsünün gittiği yaprak düğümlerdeki *d* referans görüntüsünün skorlama hesaplamasında w_k^d ağırlık değeri hesaba katılmaktadır ve denklem 4.14'de formüle edilmiştir. 4.14 formülünden de görüleceği gibi veri tabanı oluşturulurken sorgu ile eşleme skoru hesaplanan *d* referans görüntüsünün geçtiği düğümlerdeki öznitelik sayılarının, o düğüme kadarki kümülatif toplamları bir ağırlık değeri olarak kullanılmaktadır.

$$w_k^d = \sqrt{\frac{1}{\sum_{i=0}^{k} n^d(v_i)}} \tag{4.14}$$

Sorgu görüntüsünün yerleştiği her *i*. yaprak düğümdeki *d* refernas görüntüsüne iat öznitelik sayısı veya frekansı $n^d(v_i)$ şeklinde ifade edilir.

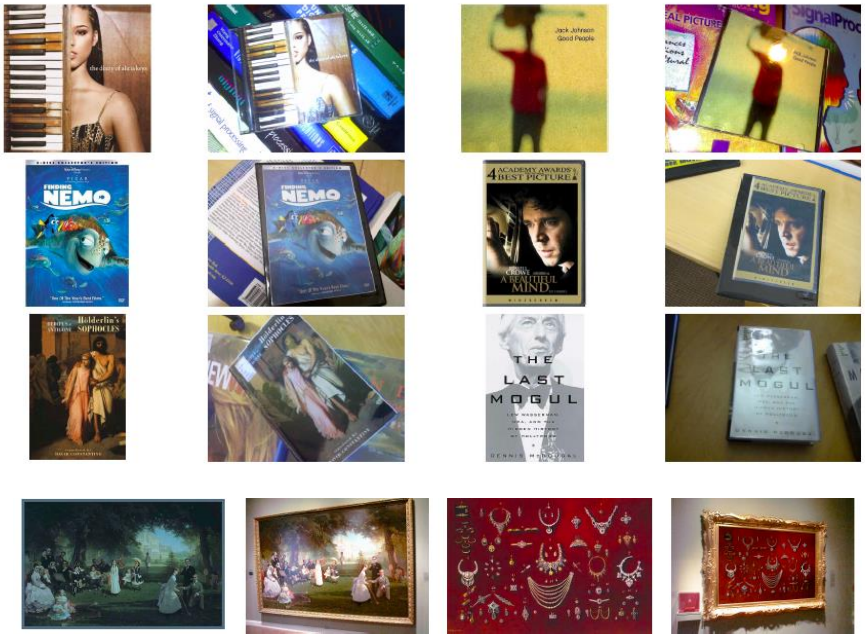
Skorlama hesaplamasında, q sorgu görüntüsü ve d referans görüntüsüne ait ortak düğümlerde $S_{q,d}(V_k)$ değeri hesaplandıktan sonra toplanarak d görüntüsü için bir S_d skorlama değeri oluşur. Yaprak düğümlerdeki tüm d referans görüntüleri ,çin oluşan skorlama değerlerinden en yüksek olanı en başarılı eşleme olarak seçilmiş olur.

Önerilen yöntemde yaprak düğümlerde hesaplanan w_k^q ve w_k^d ağırlık değerlerinin sırayla k. düğüme kadar kümülatif toplam şekilde hesaplanması, sorgu ve referans görüntülerinin geometrik benzerliklerin yakalanmasındaki ana fikri oluşturmaktadır.

5. BAŞARIM TESTLERİ VE SONUÇLAR

Görüntü özniteliklerinden ağaç oluşturmak için veri tabanında tutulacak özniteliklerin indekslenmesi, indekslenen özniteliklerden ağaç hiyerarşisinin belirlenmesi, ve veri tabanında tutulacak ağaç yapısının oluşturulması gerekmektedir.

Çalışma kapsamında veri tabanı oluşturmak için kullanılan referans görüntüler "Stanford Mobile Visual Dataset" test veri tabanında kayıtlı CD, DVD, kitap kapağı ve sanatsal tabloların orijinal görüntüleri olmak üzere toplam 100 görüntü kullanılmıştır.



Şekil 5.1 : "Stanford Mobile Visual Dataset" veri seti (Chandrasekhar, 2011)

5.1.1 Veri tabanında tutulacak özniteliklerin indekslenmesi

Veri tabanında tutulacak özniteliklerin indeklenmesi için aşağıdaki adımlar uygulanmıştır.

- Tamsayı öznitelikler çıkarıldıktan sonra bir klasöre kaydedilir.
- wxHKJar.exe çalıştırılır. "Keypoints Folder" metin kutusuna özniteliklerin

bulunduğu klasöre yazılır. Örneğin bu DVD içerisindeki ilgili dizin: "Örnek dosyalar/öznitelikler – wxHKJar girişi" şeklindedir.

- "Keypoint File Extension" kutusuna öznitelik dosyalarının uzantısı *.uzantı şeklinde yazılır. Varsayılan değeri *.key'dir.
- "Results folder" kutusuna üretilen dosyaların kaydedileceği klasör yazılır.
- Run butonuna basılır.
- Belirtilen klasörde data.txt ve descriptors.txt adlı iki dosya üretilmiş olmalıdır. wxHKJar kapatılabilir. Üretilecek dosyalara ilişkin örnekler ilgili dizinde bulunabilir: "Örnek dosyalar/wxHKJar ile üretilen".
- Çalışma kapsamında bu program Windows7 64 bit işletim sistemi üzerinde wxWidgets 2.8.12 kütüphanesi kullanılarak C++ diliyle yazılmıştır ve bu ortamda denenmiştir. Programın çalışabilmesi için wxHKJar.exe dosyasının bulunduğu dizinde libstdc++-6.dll ve libgcc_s_dw2-1.dll dosyaları da bulunmalıdır.

5.1.2 İndekslenen özniteliklerden ağaç hiyerarşisinin belirlenmesi

İndekslenen özniteliklerden ağaç hiyerarşisinin belirlenmesi için aşağıdaki adımlar uygulanmıştır.

- H-k-means.jar programı çalıştırılır.
- File menüsünden "Load descriptors file without class file" seçilir.
- wxHKJar programıyla üretilen dosyalardan descriptors.txt seçilir.
- Açılan pencereden "Set k" metin kutusuna 3, "Set a threshold" kutusuna 6 yazılır ve OK butonu tıklanır.
- Status başlıklı, "Cluster nabge schlossen" yazan küçük pencere açılana kadar beklenir.
- Ana penceredeki File menüsünden "Save clusters" seçilir ve daha önce üretilen dosyaların kaydedildiği klasöre "clusters" adıyla, uzantısız olarak bu dosya kaydedilir.
- Aynı menüden "Save virtual means" seçilir ve "virtual means" adıyla bu dosya kaydedilir.
- Aynı menüden "Save all virtual means" seçilir ve "all virtual means" adıyla bu dosya kaydedilir. H-k-means.jar programı kapatılabilir.
- Çalışma kapsamında bu program JRE6 kurulu olan Windows7 64 bit işletim

sistemi üzerinde denenmiştir.

5.1.3 Veri tabanında tutulacak ağaç yapısının oluşturulması

Veri tabanında tutulacak ağaç yapısının oluşturulması için aşağıdaki adımlar uygulanmıştır.

- hkJarMerge.exe programının bulunduğu klasöre önceki adımlarda üretilmiş 5 dosya kopyalanır.
- hkJarMerge çalıştırılmadan önce dosyaların önceki adımlarda belirtildiği şekilde adlandırılmış olduğu kontrol edilmelidir; çünkü program ancak bu dosya isimlerini görmektedir.
- hkJarMerge.exe çalıştırılır. Üzerinde yazılar akan bir konsol ekranı açılır.
- Konsol ekranı kapanana kadar beklenmelidir. Kapandığında ClustersAndMeans.txt, CM.txt ve MERGED.txt adlı üç dosya kaydedilmiş olmalıdır.
- hkJarMerge.exe ile oluşturulan MERGED.txt ve wxHKJar tarafından oluşturulan data.txt ImageMatching isimli JAVA uygulaması tarafından ağaç yapısındaki veri tabanı oluşturulurken kullanılacaktır.
- Çalışma kapsamında bu program C++ diliyle yazılmış ve Windows7 64 bit işletim sistemi üzerinde denenmiştir. Programın çalışabilmesi için hkJarMerge.exe dosyasının bulunduğu dizinde libstdc++-6.dll ve libgcc_s_dw2-1.dll dosyaları da bulunmalıdır.

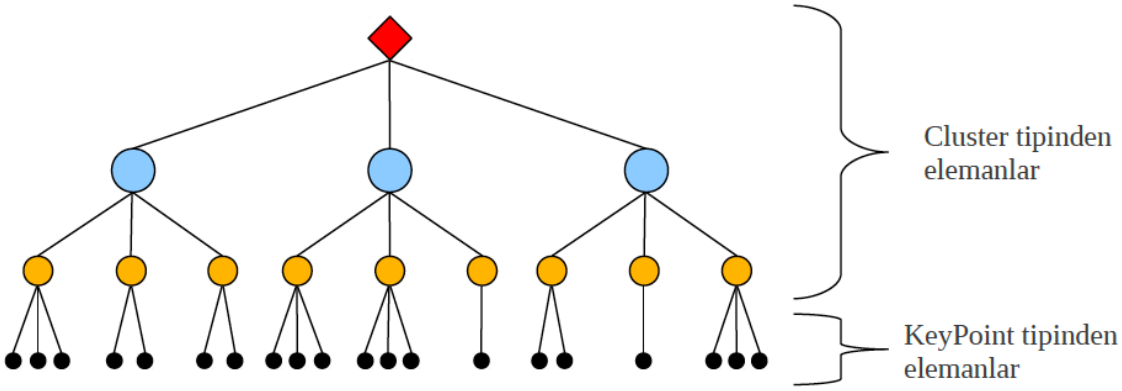
5.1.4 Veri tabanı oluşturma

Ağaç yapısında veri tabanı oluşturma için farklı veri tabanı mimarileri kullanılabilir. RDBMS (Relational Database Management System) mimarisi ile tasarlanan veri tabanları ağaç yapısında verileri için ideal veri tabanları değildir Ağaç yapısı, doğası gereği iç içe dallanan ve farklı nesne tiplerini içeren bir yapıdır. Nesne tabanlı veri tabanı mimarilerinin, ağaç yapısındaki verileri doğrudan temsil etmeye uygun olması ve üzerinde gerçekleştirilen sorgu işlemlerinin çoğul ortama uygun olması sebebiyle veri tabanının bu yapı ile oluşturulmasına karar verilmiştir.

Bir çok nesne tabanlı veri tabanı mevcuttur. Çalışmamızda açık kaynak kodlu ve Java tabanlı db4o nesne tabanlı veri tabanı kullanılmıştır.

Tez çalışması kapamında k-means hiyerarşik sınıflandırma yapılarak oluşturulan sözcük ağacı yapısındaki veri tabanında öbekleme değeri olan k = 3, derinlik değeri olan d değeri de 3 olarak alınmıştır ve veri tabanı bu şekilde oluşturulmuştur.

ImageMatching sınıfı içerisinde veri tabanı yaratmak için "create database" fonksiyonu oluşturulmuştur. Düğüm ortalama vektörleri ve görüntü öznitelikleri için sırasıyla "KeyPoint" ve "Cluster" sınıflarında veritipleri oluşturulmuştur.



Şekil 5.2 : Cluster ve KeyPoint sınıflarının ağaç yapısında görsel gösterimi

Bu bölümde sözcük ağacı (Nister, 2006), ters indeks sıkıştırma (Girod, 2011), bağlamsal ağırlıklandırma (Wang, 2011) ve önerilen arama ve eşleme metotları kullanılarak elde edilen sonuçlardan bahsedilmektedir.

5.2 Veri tabanı Üzerinde Görüntü Dağılımı

Referans olarak kullanılan "Stanford Mobile Visual Search" test veri tabanındaki CD, DVD, kitap kapağı ve sanatsal tabloların orijinal görüntülerinin oluşturulan dört farklı veri tabanı için sözcük ağacı üzerindeki düğümlere dağılımı Çizelge 5.1 de belirtilmektedir.

Çizelge 5.1 : Veri tabanlarındaki referans görüntülerin dağılımına istinaden oluşan katman, öznitelik düğüm ve yaprak düğüm sayıları

Veri Seti	**#Seviye**	**#Öznitelik**	**#Düğüm**	**#Yaprak**
Kitap	17	122555	146779	97853
CD	19	176395	211412	140911
DVD	19	158451	189844	126563
Sanatsal Tablo	19	324673	388168	258779

Çalışmada bahsi geçen dört farklı metot için gerekli olan bazı bilgiler, hesaplama karmaşıklığını azaltmak amacıyla veri tabanı oluşturulurken hesaplanmaktadır. Her bir düğüm için o düğümden geçen öznitelikleri bir öznitelik numarası ile ve özniteliklerin hangi referans görüntüye ait olduğu bilgisi veri tabanında saklanmaktadır. Ayrıca her bir düğüm için, o düğüme ait kök düğümden başlayan bir indeks numarası bulunmaktadır. Kök düğümün indeks numarası 0'dır. Kök düğümden itibaren sırasıyla dallanma faktörünü 3 olacak şekilde dallanarak bir alt dalda indeks değeri 1,2 ve 3 olacak şekilde düğümler oluşur. Bu dallanma ile beraber ağaç yapısında seviyeler oluşur. Kullanılan veri setleri için maksimum seviyeler Çizelge 5.1 de görülmektedir. Düğümler için seviye bilgisini de gösterecek şekilde "clusterid" adını verdiğimiz "cluster_13_2500" benzeri bir indeks değeri de tutulmaktadır. Bu indeks değerindeki 13 seviye bilgisini, 2500 ise o seviyedeki düğüm sıralamasını ifade etmektedir.

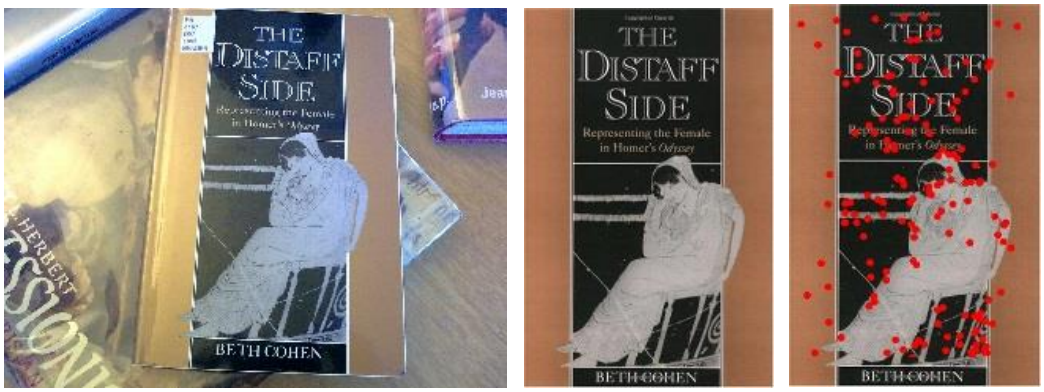
Şekil 5.3 de görülen Nokia 5800 telefonla çekilmiş bir kitap sorgu ve referans görüntüsü örneği görülmektedir. En sağdaki görüntüde referans görüntüsünün öznitelikleri görülmektedir.

Sorgu görüntüsü ait 2666 adet yaprak düğüm bulunmaktadır. Bu yaprak düğümlerin seviye bazlı dağılım Çizelge 5.2 deki gibidir.

Çizelge 5.2 : Nokia 5800 ile çekilen sorgu görüntüsünün seviye bazlı yaprak düğüm dağılımı

Seviye (Layer)	Sayı
7	1
8	15
9	118
10	491
11	850
12	666
13	323
14	133
15	54
16	12
17	2

Çizelge 5.2 e baktığımız açık bir şekilde en çok yaprak düğüm dağılımının 10 - 12 seviyelerin olduğu görülmektedir. Referans görüntü için de benzer şekilde Çizelge 5.3 seviye bazlı yaprak düğüm dağılımı görülmektedir. Sözcük ağacında referans görüntüsüne ait 895 yaprak düğüm bulunmaktadır.



Şekil 5.3 : Nokia 5800 telefonla çekilmiş bir kitap sorgu ve referans görüntüsü

Çizelge 5.3 : Nokia 5800 ile çekilen referans görüntüsünün seviye bazlı yaprak düğüm dağılımı

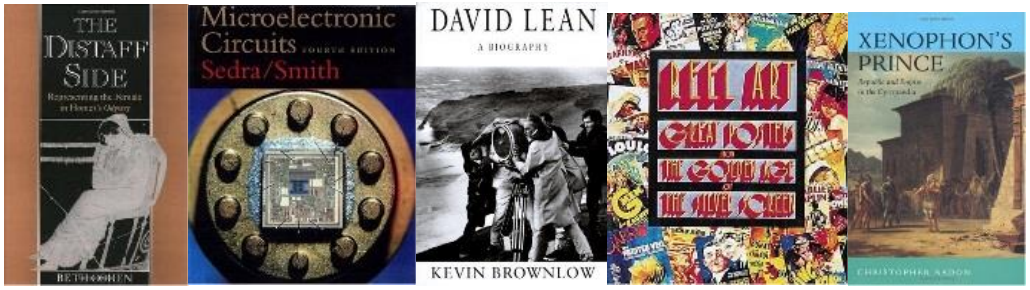
Seviye (Layer)	Sayı
8	9
9	48
10	205
11	285
12	200
13	96
14	33
15	16
16	2

Çizelge 5.4'de sorgu görüntüsünün en başarılı şekilde eşlendiği beş referans görüntüye ait bilgiler görülmektedir. Referans görüntülerin görselleri Şekil 5.4 de bulunmaktadır. Skor değeri en yüksek olan görüntü en iyi eşlene görüntüdür. Çizelge 5.4 de 001.jpg referans görüntüsü Şekil 5.3 deki referans görüntüdür. Sorgu görüntüsü sözcük ağacına yerleştirildikten sonra özniteliklerinin yerleştiği yaprak düğümlerdeki referans görüntülere baktığımızda bu beş görüntünün ortak yaprak düğüm sayısı belirtilmiştir. En yüksek skora sahip görüntü en fazla ortak düğüm sayısına sahip görüntüdür. Fakat sıralamaya baktığımızda skor değer ile benzer bir korelasyon olmadığı görülüyor. 050.jpg görüntüsü veri tabanındaki en fazla özniteliği olan referans görüntüsü. Skorlamaların geneline bakıldığına fazla sayıda özniteliğinden dolayı yanlış eşlemeler 050.jpg referans görüntüsü ile olmaktadır. Şekil 5.4 de Nokia 5800 telefon ile çekilen sorgu görüntüsü ve en iyi eşlenen beş referans görüntü görülmektedir.

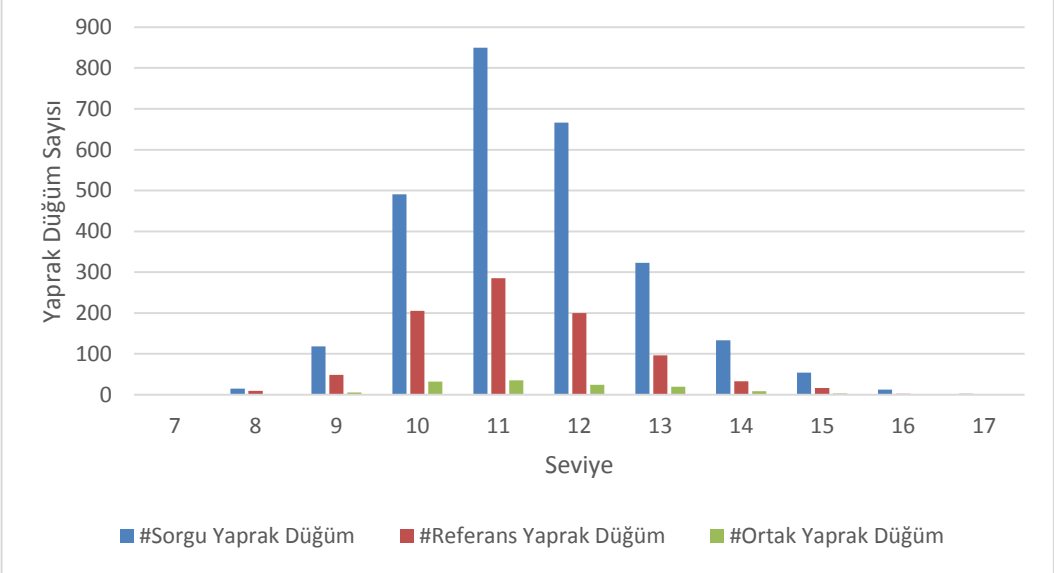
Çizelge 5.4 : Nokia 5800 ile çekilen sorgu görüntüsü ile en iyi eşlene beş görüntü

Sorgu Görüntüsü	#Sorgu Görüntüsü Yapsak Düğüm Sayısı	Eşlenen Referans Görüntüsü	#Referans Görüntü Yaprak Düğümleri	#Ortak Yaprak Düğüm Sayısı	Skor Değeri
5800_001	2666	001.jpg	894	127	0.241
5800_001	2666	087.jpg	2264	87	0.098
5800_001	2666	013.jpg	1634	73	0.096
5800_001	2666	050.jpg	4700	116	0.092
5800_001	2666	021.jpg	1508	62	0.088

Şekil 5.5 da sorgu ve en iyi eşlenen 001.jpg referans görüntüsü için seviye bazlı yaprak düğüm ve ortak yaprak düğümlerin dağılım karşılaştırması görülmektedir.



Şekil 5.4 : Nokia 5800 telefonla çekilmiş en iyi eşlenen beş görüntü.



Şekil 5.5 : Sorgu ve 001.jpg referans görüntüsü yaprak düğüm dağılımı

5.3 Görüntü Eşleme Başarımı Uygulama Alanı

Gerçeklenen mobil ürün eşleme sisteminin başarımı, "Stanford Mobile Visual Search" test veri tabanında kayıtlı ve E63, Nokia 5800, Droid, Palm ve iPhone gibi beş farklı cep telefonu ile farklı çözünürlüklerde çekilmiş görüntüler kullanılarak test edilmiştir.

Bu bölümde, SIFT öznitelikleri ile modellenerek sözcük ağacı üzerinde indekslenen mobil ürün eşleme sisteminin (Nister, 2006) başarım sonuçları, ters indeks (Girod, 2011) ve bağlamsal ağırlıklandırma (contextual weighting) (Wang, 2011) uygulanarak elde edilen sonuçlarla karşılaştırılarak yorumlanmıştır. Çizelge 5.5, sözcük ağacı (Nister, 2006) kullanılarak elde edilen mobil ürün eşleme sonuçlarını gösterirken, Çizelge 5.6 sözcük ağacı üzerinde ters indeks (Girod, 2011) uygulanarak elde edilmiş sonuçları gösterir. Bunun yanı sıra Çizelge 5.7 de ise sözcük ağacının bağlamsal ağırlıklandırma (Wang, 2011) uygulanması ile elde edilen sonuçlara yer verilmiştir.

Çizelge 5.5 : Sözcük ağacı eşleme başarım sonuçları.

	Sözcük Ağacı	**1**	**2**	**3**	**4**	**5**	**6**	**10**	**15**
CD	E63	43	52	55	56	58	61	68	72
	Palm	16	26	29	35	39	40	48	59
	Droid	58	69	75	76	78	81	82	83
Book	iPhone	90	95	96	97	97	97	98	99
	5800	61	70	77	82	86	87	92	94
	Droid	74	81	88	92	94	94	95	96
DVD	E63	38	54	65	74	78	81	85	88
	Palm	40	63	71	75	78	80	86	88
	Droid	34	45	55	59	61	68	74	79
Tablo (Paint)	E63	20	27	30	34	38	41	51	58
	Palm	22	29	34	35	39	41	50	55
	Droid	23	31	38	43	47	52	61	69
Ortalama		43,3	53,50	59,42	63,17	66,08	68,58	74,17	78,33

Çizelge 5.5 ve Çizelge 5.6 de yer alan sonuçlardan da gözlenebildiği üzere, sözcük ağacı üzerinde ters indeks kullanılması başarım sonuçlarını arttırmaktadır. Eşleme başarı yüzdeleri göz önüne alındığında, ters indeks yöntemi tüm kategori ve cihazlarda ilk seferde eşleme başarı oranını %43'den %45'e çıkarken, deneme sayısı arttıkça başarım oranları arasındaki açı da artmıştır.

Çizelge 5.6 : Ters indeks eşleme başarım sonuçları.

	Ters İndeks	**1**	**2**	**3**	**4**	**5**	**6**	**10**	**15**
CD	E63	52	63	71	73	76	77	82	89
	Palm	27	38	45	55	59	65	77	84
	Droid	58	62	66	69	74	76	86	91
Book	iPhone	80	92	95	95	96	97	98	99
	5800	56	71	77	82	87	88	94	97
	Droid	65	79	84	86	92	94	97	99
DVD	E63	51	66	77	83	86	87	92	97
	Palm	53	64	76	77	82	84	88	91
	Droid	36	41	45	50	58	60	69	73
Tablo (Paint)	E63	22	34	36	40	44	46	57	67
	Palm	17	25	26	35	37	38	47	60
	Droid	25	34	40	43	48	52	64	73
Ortalama		45	55	61	65	70	72	79	85

Yapılan testler sırasında ters indeks kullanılmış sistemin başarımı arttırdığı sonucu elde edilmiştir (Çizelge 5.6). Bunun yanı sıra model üzerinde ters indeksleme yapılmasının eşleme süresinde de artışa yol açtığı gözlenmiştir. Bunun sebebi ise ters indeksin geniş bellek kullanımının sunucu üzerindeki işlemleri yavaşlatmasından kaynaklanmaktadır. Süredeki artışın sistem performansı üzerinde ciddi bir etkisi olmaması ve başarım oranındaki artış değerlendirilerek ters indeksin mobil ürün eşleme sistemini geliştirici, ilerletici bir yöntem olduğuna karar verilmiştir.

Çizelge 5.7 : Bağlamsal ağırlıklandırma eşleme başarım sonuçları.

	Bağlamsal Ağırlıklandırma	**1**	**2**	**3**	**4**	**5**	**6**	**10**	**15**
CD	E63	57	65	71	76	76	77	83	93
	Palm	31	41	45	58	61	65	76	86
	Droid	61	63	69	70	75	78	88	92
Book	iPhone	72	88	93	98	98	98	98	98
	5800	52	69	75	82	85	88	93	93
	Droid	77	89	93	95	95	96	98	99
DVD	E63	46	63	68	72	76	77	82	84
	Palm	44	69	72	77	78	79	81	86
	Droid	38	79	85	88	90	91	91	92
Tablo (Paint)	E63	41	50	52	56	58	61	62	69
	Palm	26	44	49	54	58	59	68	71
	Droid	19	48	57	58	60	62	66	69
Ortalama		47	64	69	74	76	78	82	86

Çizelge 5.7'te sözcük ağacı üzerinde bağlamsal ağırlıklandırma uygulanmış modelin başarım sonuçları yer almaktadır. Görüldüğü üzere bağlamsal ağırlıklandırma yöntemi ilk seferde yaklaşık %2'lik bir ortalama başarım artışı sağlamıştır.

Bağlamsal ağırlıklandırma yaklaşımında sorgu ve referans görüntülerinin her bir öznitelik vektörü için ağaçta izlediği yol bilgisi tutulmaktadır. Yol bilgisi tutulmasıda ekstra bir verinin daha bellekte tutulması anlamına gelmektedir. Ayrıca diğer yöntemlerde düğüm bazında işlem yapılırken bu yaklaşımda öznitelik bazında işlem yapılmaktadır ve bu da hesaplama karmaşıklığını ve eşleme getirme zamanını arttırmaktadır.

Sözcük ağacı üzerinde ters indeks sıkıştırma ve bağlamsal ağırlıklandırma methodlarının uygulanması başarım oranını arttırırken, bu artış istenilen ölçüde olmamıştır.Özellikle ilk seferde yapılan eşleşmelerde elde edilen başarının daha da yükseltilmesi için daha özgün bir yaklaşımın geliştirilmesi gerektiğine karar verişmiştir.

Çizelge 5.8 : Önerilen Yaprak Düğüm Ağırlıklandırma Eşleme Başarım Sonuçları.

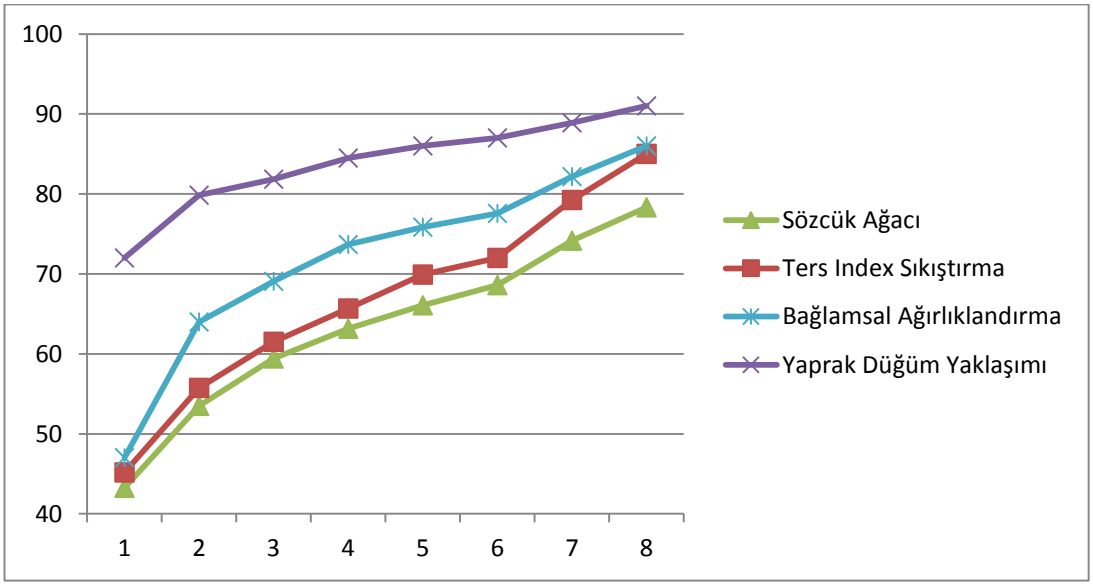
	Yaprak Düğüm Ağırlıklandırma	**1**	**2**	**3**	**4**	**5**	**6**	**10**	**15**
CD	E63	76	87	90	93	93	96	96	97
	Palm	48	60	61	62	67	69	72	75
	Droid	76	86	87	88	90	92	92	96
Book	iPhone	98	99	99	99	99	99	99	99
	5800	81	87	90	92	93	94	96	99
	Droid	94	97	98	98	98	98	99	99
DVD	E63	73	84	86	94	95	96	97	98
	Palm	79	89	90	93	93	93	96	97
	Droid	80	87	90	93	93	93	96	97
Tablo (Paint)	E63	56	65	66	70	72	73	75	78
	Palm	47	57	61	62	68	69	72	73
	Droid	56	60	64	70	71	72	77	79
Ortalama		72	79,83	81,83	84,5	86	87	88,91	91

Bu tez çalışmasında sunulan ve özgün bir yaklaşım olan yaprak düğümağırlıklandırma yönteminin sözcük ağacı uygulanmasının ardından elde edilen başarım sonuçları

Çizelge 5.8'te yer almaktadır. Bu yöntem ortalama başarım oranını ilk seferde %72'ye kadar çıkarmıştır.

Çizelge 5.5, 5.6, 5.7 ve 5.8 'deki verilerin karşılaştırmaları Şekil 5.6'de gösterilmiştir. Şekil 5.6'dan da gözlenebileceği gibi, yaprak düğüm ağırlıklandırma uygulanarak geliştirilmiş mobil ürün eşleme sisteminde en başarılı sonuçlar elde edilmiştir. Özellikle ilk seferde eşleşme başarımlarında ortalama başarım oranını %72'ye kadar yükselten bu yöntem, 15. denemede eşleşme başarımını %90 oranının üstüne çıkarmıştır. Bunun en önemli sebeplerinden biri ise önerilen çalışmada sorgu görüntüsünün yerleştiği yaprak düğümler üzerinde işlem yapılarak, bu yaprak düğümlerdeki sorgu ve referans görüntüleri için ağrılık değerlerinin hesaba katılmasıdır.

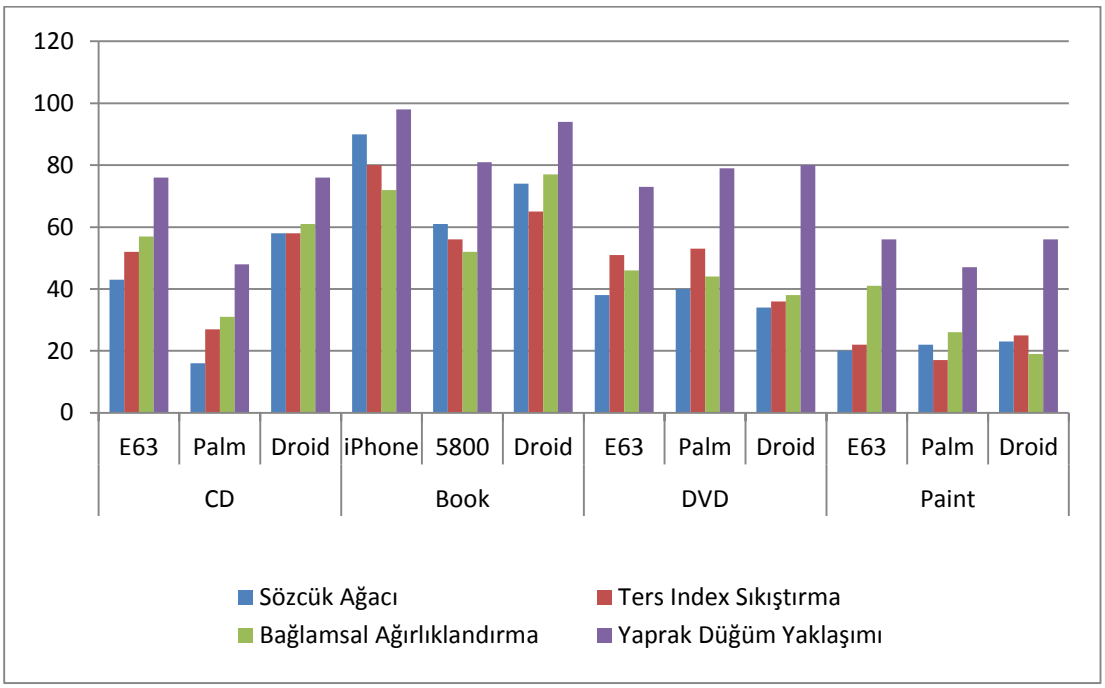
Önerilen yöntemde yaprak düğümlerde hesaplanan ağırlık değerlerinin kümülatif toplam şekilde hesaplanması, sorgu ve referans görüntülerinin geometrik benzerliklerin yakalanmasındaki ana fikri oluşturmaktadır ve eşleme başarımını arttırdığı görülmektedir.



Şekil 5.6 : Bahsedilen dört yöntem için başarım grafiği

Şekil 5.7, tüm kategori ve cihazlardan elde edilen başarım sonuçları ortalamalarının ayrıntılı karşılaştırmalarını sütun grafiği olarak göstermektedir. Grafikten gözlendiği üzere mor renk sütunlarla ifade edilen ve sözcük ağacı üzerindekök-tabanlı ağırlıklandırmanın uygulandığı sistemin başarımlarını gösteren sonuçlar, tüm kategori

ve cihazlarda bu yöntemin diğer yöntemlerden yüksek başarım elde ettiğini kanıtlamaktadır.



Şekil 5.7 : Bahsedilen dört yöntem için başarım grafiğinin sütun hali

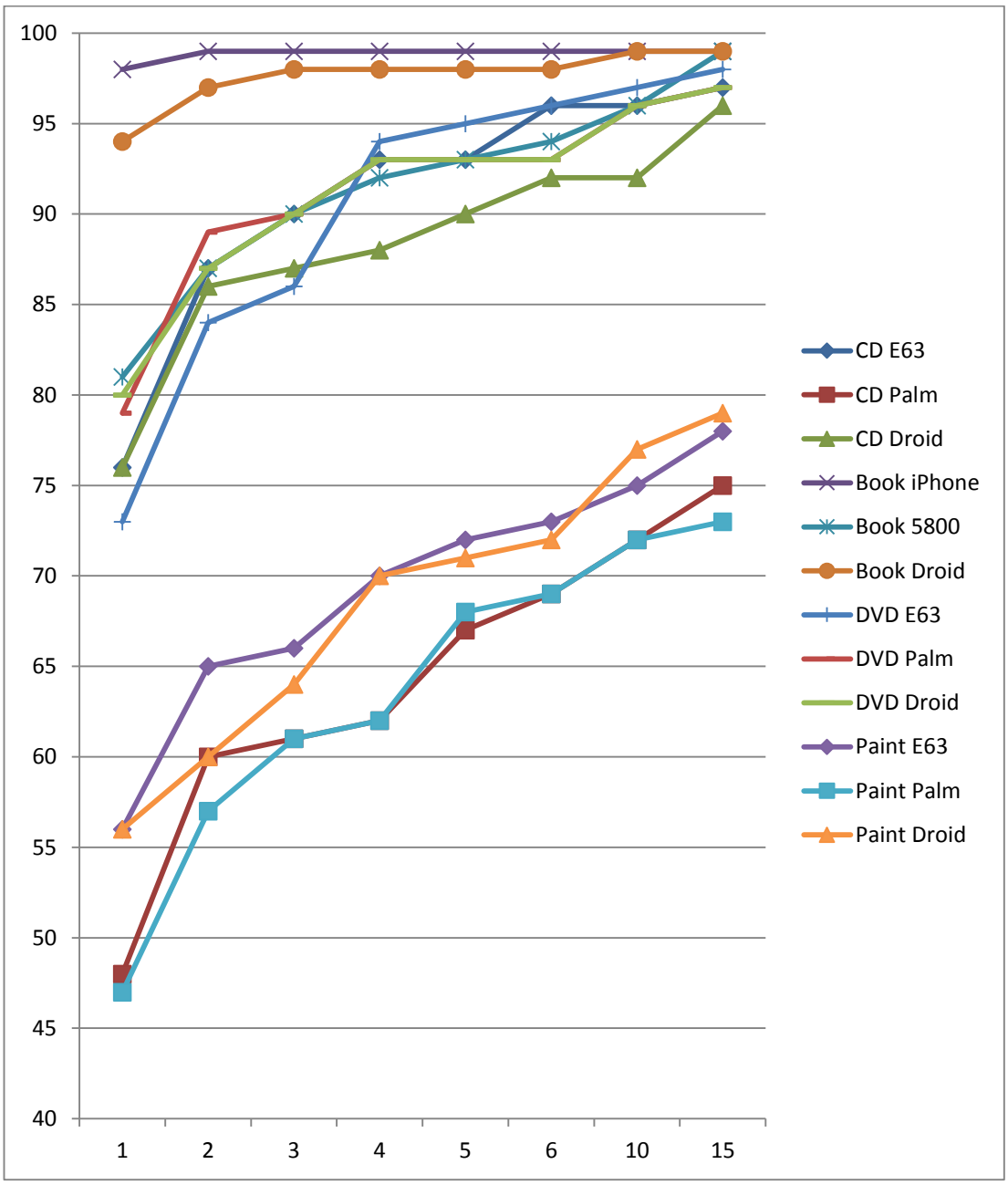
5.4 Önerilen Arama ve Eşleme Yöntemi Sonuçlarının Detaylı Yorumu

Bir önceki bölümde farklı metotlar kullanılarak elde edilen başarım sonuçları karşılaştırılmıştır. Bu bölümde ise en yüksek başarım oranının gözlendiği, önerilen sorgu görüntüsünün yaprak düğümleri üzerinde ağırlıklandırma uygulanması ile elde edilen sonuçların kategori ve cihaz bazlı sonuçları ayrıntılı olarak incelenmiştir. CD, kitap, DVD ve tablodan oluşan dört kategoride E63, Nokia 5800, Droid, Palm ve iPhone gibi beş farklı cep telefonu ile çekilmiş görüntülerden elde edilen sonuçların başarım oranları Şekil 5.8'te gösterilmiştir.

Kategori bazında karşılaştırma yapmak için dört kategori için de ortak cihaz olan Droid seçildiğinde, ilk seferde eşleşmede en yüksek başarımın kitap kapaklarında (%94) en düşük başarımın ise tablolarda (%56) elde edildiği gözlenmiştir. Bunun yanı sıra tüm cihaz türleri arasında karşılaştırma yapıldığında, kitap kapakları üzerinde iPhone cihazı kullanılarak yapılan testlerde sadece ilk seferde %98 oranında başarı elde edilmiştir.

Sonuçlarda dikkat çeken başka bir özellik ise kategori bazlı elde edilen ortalama başarılarla cihaz bazlı değerlendirmeler arasında doğru orantı olmamasıdır. Örneğin,

CD ve tablo kategorisinde E63 ile çekilmiş görüntüler Palm ile çekilenlerden daha yüksek başarım sonucu verirken, DVD kategorisinde Palm ile çekilen görüntüler daha yüksek oran elde etmiştir. Elde edilen bu gözlem, başarım oranlarının sadece cihaz çözünürlükleri ile ölçülemeyeceğini, SIFT algoritmasından çıkarılan yerel özniteliklerin referans görüntülerden elde edilenlerle örtüşme seviyesinin doğrudan çözünürlüğe bağlı olmadığını gözler önüne sermiştir.



Şekil 5.8 : Farklı cihazlar ile farklı nesnelerin eşleme başarım oranı

6. SONUÇLAR

Mobil cihazlar kullanılarak yüksek sayıda görüntü içeren büyük görüntü veri tabanlarında görüntü arama ve eşleme ve geri getirme problemi gerek akademik çalışmalarda gerekse de güncel uygulamalar açısından önemli bir araştırma alanıdır. Mobil bir cihaz tarafından çekilen bir görüntünün anlık olarak bir veri tabanından sorgulanması ve anlamlandırılması konusunda birçok güncel mobil uygulama bulunmaktadır. Sanal gerçeklik uygulamaları olarak da adlandırılan, mobil cihazın kamerası kullanarak bir logo tanıma, bir CD kapağının fiyatını sorgulama ve bilgi getirme veya bir nesneyi tanıyarak hakkında bilgi sahibi olma gibi uygulamalar son dönem çoğul ortam sistemlerinde sıklıkla kullanılmaktadır. Bunun yanı sıra sağlık alanında, medikal görüntü eşleme de bir başka kullanım alanıdır. Bir hastanın çekilen beyin tomografisindeki değişimlerin belirlenmesi ve otomatik teşhis konulması gibi çalışmalar devam etmektedir.

Görüntü tabanlı sorgulama amaçlı kullanılan arama motorları halen üzerinde aktif çalışılan bir alandır. Arama motorları üzerinden sorgulanan bir görüntünün en benzer görüntüler ile eşleştirilmesi veya görüntüye ilişkin bilgilerin getirilmesi gibi uygulamalarda hızlı ve yüksek doğrulukla çalışan arama motorlarının geliştirilmesi önemlidir. Ayrıca mobil ve web pazarlama amacı ile de sorgulanan görüntü ile ilintilendirilebilecek bilgilerin, fiyat bilgisi, en ucuz fiyatı veren e-ticaret sitelerinin bulunması gibi alınabilmesi de önemli bir kullanım alanıdır.

Video içerisinde görüntü arama bir başka kullanım alanı olarak ele alınmıştır. Mevcuttaki video içerik sağlayıcı siteler de arama faaliyetleri daha çok metin tabanlı videoyu tanımlayan kısa içerikler üzerinden yapılmaktadır. Video sahneleri içerisinden bir nesneyi tanıma veya anahtar-görüntülere dayalı olarak benzer sahneleri getirme gibi alanlarda yeni uygulama alanlarıdır. Bu sayede izinsiz korsan yayınlanan film ve dizilerin sanal ortamda bulunması da sağlanabilmektedir.

Çoklu ekran uygulamaları gibi yeni kullanım alanları da söz konusudur. İçeriklerin mobil cep telefonu için ayrı, tabletler için ayrı veya televizyon gibi büyük ekranlar için

ayrı bir şekilde gösterilmesi ve bu içeriklerin platformlar arası geçişlerde süreklilik sağlanarak aktarımının devam etmesi konusu bir diğer önemli uygulama alanıdır.

Bu tez çalışmasında, literatürdeki çeşitli görüntü eşleme ve getirme yöntem ve yaklaşımları kullanılarak yüksek başarımlı eşleme performansına sahip bir mobil görüntü arama/eşleme sistemi geliştirmek ve kullanılan bu yaklaşımların eşleme başarımına etkilerinin gözlemlenmesi hedeflenmiştir. Yerel SIFT özniteliklerinin geleneksel geometrik uzaklık tespiti ile eşlenmesine ek olarak, özniteliklerin sözcük ağacı üzerinde indekslenmesi, ters indeks kullanımı ve ağaç üzerinde bağlamsal ağırlıklandırma uygulanması gibi farklı yaklaşımlarla modelleme yapılarak analiz gerçekleştirilmiştir.

Bu çalışmada elde edilen sonuçlara göre; sözcük ağacı, ters indeks ve bağlamsal ağırlıklandırma metotlarına istinaden önerilen yaprak düğüm yöntemi arama/eşleme sisteminin eşleme başarımını arttırmaktadır. Varolan sistemlerle karşılaştırmalı olarak yapılan başarım testlerinden elde edilen veriler, özellikle dördüncü senaryo sonuçları, önerilen arama/eşleme yönteminin arama performansını arttırarak eşleme başarımında diğer durumlara oranla ciddi oranda artış sağladığını göstermektedir. Sözcük ağacı yöntemi ile ortalama eşleme başarımı %43, ters indeks eşleme ile %45, bağlamsal ağırlıklandırma ile %47 olarak ölçülürken önerilen yaprak düğüm tabanlı eşleme ile %72 doğru eşleme başarımına ulaşılmıştır.

Önerilen çalışmaya ek olarak SIFT özniteliklerinde bulunan ışıklılık, ölçek, rotasyon, öznitelik yer bilgisi gibi parametrelerin de skorlamaya eklenmesiyle eşleme başarımı arttırılabilir. Yapılan çalışmalarda eşleme başarımı dışında performans ölçüm kriterleri de değerlendirilebilir. Benzer referans görüntü seti kullanılarak farklı öbekleme parameteleri ile test sonuçlarındaki eşleme başarımı değerlendirilebilir. Farklı öbekleme parametreleri kullanılması veri tabanın yeniden oluşturma ve indeksleme yapmak anlamına da gelmektedir. Teorik olarak bir yenilik ekleme açısından benzer öznitelikler için kümeleme yapılması ve daha az sözcük ağacı dalları üzerinde arama sağlanması düşünülebilir. Böylece performans artışı ve eşleme başarımı artışı sağlanabilir. Mevcut arama eşleme çalışmasında sözcük ağacında az ziyaret edilen dalların belirlenerek skorlamaya dahil edilmemesi de düşünülebilir.

Bir başka bakılabilecek alan da SIFT özniteliklerinin ikili (binary) olarak tutulması olabilir. Bu şekilde arama hızı artacaktır fakat veri tabanı oluşturma kısmı yeni bir tasarım gerektirmektedir.

KAYNAKLAR

Baldock, R. (1992). Trainable models for the interpretation of biomedical images. *Image and Vision Computing*, 10(6), pp.444-449.

Black, M. and Jepson, A. (1998). Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 1(26), pp.63-84.

Chandrasekhar, V., Bach, J., Girod, B., Chen, D., Tsai, S., Cheung, N., Chen, H., Takacs, G., Reznik, Y., Vedantham, R. and Grzeszczuk, R. (2011). The stanford mobile visual search data set. *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11*.

CNET, (2009). *Nokia's Point & Find gives new life to the camera phone - CNET*. [online] Available at: http://www.cnet.com/news/nokias-point-find-gives-new-life-to-the-camera-phone/ [Accessed 16 Nov. 2015].

Elmasri and Navathe, (2011). Fundamentals of Dtabase Systems. US. Pearson Addison-Wesley

Girod, B., Chandrasekhar, V., Grzeszczuk, R. and Reznik, Y. (2011). Mobile Visual Search: Architectures, Technologies, and the Emerging MPEG Standard. *IEEE Multimedia*, 18(3), pp.86-94.

Google.com, (2015). *Google Mobile*. [online] Available at: http://www.google.com/mobile/ [Accessed 16 Nov. 2015].

Guerrero, M. (2011). *A Comparative Study of Three Image Matcing Algorithms: Sift, Surf, and Fast.*(Ms Thesis). Utah State University

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In: *the 4th Alvey Vision Conference*. pp.147 - 150.

Hopkins, B. (2015). *Google Goggles*. [online] PCWorld. Available at: http://www.pcworld.com/article/202213/google_goggles_android_app.html [Accessed 16 Nov. 2015].

Joshi, H. and Sinha, K. (2013). A survey on image mosaicing techniques. *International Journal of Advanced Research in Computer Engineering &*

Technology (IJARCET), 2(2).

Junior, J., Marşal, R. and Batista, M. (2014). Image Retrieval: Importance and Applications. In: *X Workshop de Vis Ìfao Computacional - WVC 2014*.

Kooaba.com, (2015). *kooaba - World's leading platform for image recognition | kooaba*. [online] Available at: http://kooaba.com [Accessed 16 Nov. 2015].

Leonardis, A. and Bischof, H. (2000). Robust Recognition Using Eigenimages. *Computer Vision and Image Understanding*, 78(1), pp.99-118.

Lindeberg, T. (1994). *Scale-space theory in computer vision*. Boston: Kluwer Academic.

Lindeberg, T. (1994). Scale-space theory: a basic tool for analyzing structures at different scales. *CJAS*, 21(1), pp.225-270.

Matas, J. and Obdrzalek, S. (2004). Object recognition methods based on transformation covariant features. In: *XII. European Signal Processing Conference EUSIPCO*. pp.1333-1336.

Mathworks.com, (2015). *Find corner points in image - MATLAB corner*. [online] Available at: http://www.mathworks.com/help/images/ref/corner.html [Accessed 13 Nov. 2015].

Mikolajczyk, K. (2002). *Detection of Local Features Invariant to Affine Transformations*. Institute National Polytechnique de Gronoble.

Nister, D. and Stewenius, H. (2006). Scalable Recognition with a Vocabulary Tree. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, pp.2161 - 2168.

Pope, A. (1994). *Model-based object recognition - a survey of recent research*.

Rao, L. (2008). *Image Recognition Startup SnapTell Acquired by Amazon Subsidiary.com*. [online] TechCrunch. Available at: http://techcrunch.com/2009/06/16/image-recognition-startup-snaptell-acquired-by-amazon-subsidiary-a9com/ [Accessed 16 Nov. 2015].

Rey Otero, I. and Delbracio, M. (2014). Anatomy of the SIFT Method. *Image Processing On Line*, 4, pp.370-396.

Robinson, G., Colchester, A. and Griffin, L. (1994). Model-based recognition of anatomical objects from medical images. *Image and Vision Computing*, 12(8), pp.499-507.

Solem, J. (2012). *Programming computer vision with Python*. Sebastopol, CA: O'Reilly.

Swain, and Ballard, (1990). Indexing Via Color Histograms. In: *Third International Conference on Computer Vision*. pp.390-393.

Trajkoviç, M. and Hedley, M. (1998). Fast corner detection. *Image and Vision Computing*, 16(2), pp.75-87.

Tsai, S., Chen, D., Chandrasekhar, V., Takacs, G., Cheung, N.M., Vedantham, R., Grzeszczuk, R., and Girod, B. (2010). Mobile product recognition. In: *Proceedings of ACM International Conference on Multimedia*, p.p.1587- 1590.

Tsai, S., Chen, D., Chandrasekhar, V., Takacs, G., Cheung, N.M., Vedantham, R., Grzeszczuk, R., and Girod, B. (2010). Fast geometric re-ranking for image based retrieval, In: *International Conference on Image Processing*, 17, pp. 1029 – 1032.

Torralba, A., Fergus, R., and Weiss, Y. (2008). Small codes and large image databases for recognition, In: *IEEE Conf. Comput. Vis. Pattern Recognit.,* Anchorage, AK, USA, pp. 1–8.

Trzcinski, T., Christoudias, M., Lepetit, V. and Fua, P. (2012). Learning image descriptors with the boosting-trick. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 37, pp.597 – 610

Yu, C., and Brandenburg, T., (2011). Multimedia Database Applications: Issues and Concerns for Classroom Teaching. In: *The International Journal of Multimedia & Its Applications (IJMA),* Vol.3, No.1, pp.1-9

Wang, X., Yang, M., Cour, T., Zhu, S., T.X., H. and Yu, K. (2011). Contextual weighting for vocabulary tree based image retrieval. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. Barcelona: IEEE, pp.209 - 216.

Wang, X., Yang, M., and Yu, K. (2011). Efficient Re-ranking in Vocabulary Tree based Image Retrieval. In: *Conference Record of the 45th Asilomar Conference on Signals, Systems and Computers*, pp. 855-859

Witkin, A. (1984). Scale-space filtering: A new approach to multi-scale description. In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*. IEEE, pp.150-153.

Xiaoguang, Hsu, and Jain, ed., (2004). Face recognition with 3D model-based synthesis. In: *Biometric Authentication*, 1st ed. Springer, pp.139--146.

Zhang, S., Cour, T., Yu, K., and Dimitris, N.M., (2015). Query Specific Rank Fusion for Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 37, pp.803 – 815

ÖZGEÇMİŞ



Ad Soyad : Onur Çalıkuş

Doğum Yeri ve Tarihi : Akhisar MANİSA 23.01.1981

E-Posta : onurcalikus@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2003, Yıldız Teknik Üniversitesi, Elektrik Elektronik Fakiltesi, Elektronik ve Haberleşme Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

2010 – 2012	Türk Telekom	Kurumsal Servisleri Mobil Ürünler Grup Müdürü
2010 – 2012	Avea	Mobil Ürünler ve Servisleri Takım Lideri
2010 – 2012	Avea	Ar-Ge Yazılım Uzmanı
2010 – 2009	Turkcell	Ürün Yöneticisi
2008 – 2009	Turkcell	Ar-Ge Yazılım Geliştirme
2004 – 2008	Vodafone	VAS Yazılım Geliştirme Uzmanı

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- Çırakman, Ö., Kutluk, S., Günsel, B., Çalıkuş, O., Mobil ürün algılama amaçlı bir sözlük ağacı gerçeklemesi. *2012 IEEE SIU*

DİĞER YAYINLAR, SUNUMLAR VE PATENTLER:

- Yüksek-Ölçekli Mobil İletişim Verilerinin Açık-kaynak Hadoop Çerçevesi Kullanılarak Paralel Ve İş-hatlı İşlenmesi, *Melih Koca, Ozyegin University; Ismail Ari, Ozyegin University; Onur Çalıkuş, Avea Labs; Cenk Sezgin, Avea; Uğur Koçak, Ozyegin University*
- An Operator Provided M-Learning Service: A Preliminary Report, *Onur Çalıkuş, Cenk Sezgin, Haluk Bingol, M. Gokhan Habiboglu, Suzan Uskudarli, Ahmet Yildirim*