KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANAHTAR NOKTASI TABANLI KOPYALA YAPIŞTIR SAHTECİLİĞİ TESPİTİ

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Gül MUZAFFER

MAYIS 2016
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANAHTAR NOKTASI TABANLI KOPYALA YAPIŞTIR SAHTECİLİĞİ TESPİTİ

Bilgisayar Müh. Gül MUZAFFER

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
" BİLGİSAYAR YÜKSEK MÜHENDİSİ "
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 12 / 05 /2016
Tezin Savunma Tarihi : 10 / 06 /2016

Tez Danışmanı : Yrd. Doç. Dr. Güzin ULUTAŞ

Trabzon 2016

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalında
Gül MUZAFFER Tarafından Hazırlanan

ANAHTAR NOKTASI TABANLI KOPYALA YAPIŞTIR SAHTECİLİĞİ TESPİTİ

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 17 / 05 / 2016 gün ve 1653 sayılı kararıyla oluşturulan jüri tarafından yapılan sınavda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Prof. Dr. Albert LEVİ .....

Üye : Prof. Dr. Vasif V. NABİYEV .....

Üye : Yrd. Doç. Dr. Güzin ULUTAŞ .....

Prof. Dr. Sadettin KORKMAZ
Enstitü Müdürü

ÖNSÖZ

Teknolojinin hızla gelişmesiyle birlikte sayısal görüntülerin kullanım alanı yaygınlaşmış olup bununla birlikte sayısal görüntülerde gerçekleştirilen sahtecilik oranı da artmıştır. Bu sahteciliklerden en yaygın olanı kopyala yapıştır sahteciliğidir.

Çalışmada literatürdeki anahtar noktası tabanlı kopyala yapıştır sahteciliği yöntemleri incelenerek tespit edilen problemleri iyileştirmek adına yeni yöntemler önerilmiştir. Önerilen yöntemler üç şekilde gerçekleştirilmiştir. İlk olarak anahtar noktası tabanlı bir yöntemin renk kanallarında uygulanmasıyla doğruluk oranı daha yüksek sonuçlar elde edilmiştir. İkinci çalışmada anahtar noktası tabanlı yöntemin özellikle düz bölgelerde başarısız olması sebebiyle görüntülerden önce doku çıkarma işlemi gerçekleştirilmiş ve daha sonra kopyala yapıştır sahteciliği tespiti yapılmıştır. Bu bakış açısı ile literatürde var olan iki yöntemin iyileştirilmesi gerçekleştirilmiştir. Son yapılan çalışmada ise doku çıkarma ön işlemini ortadan kaldırarak yine de düz bölgelerde etkin bir şekilde çalışan yeni bir kopyala yapıştır sahteciliği tespiti yöntemi önerilmiştir.

Çalışmalarımda danışmanlığımı üstlenen değerli hocam Yrd. Doç. Dr. Güzin ULUTAŞ' a ilgi, destek ve tecrübelerinden dolayı teşekkürlerimi borç bilirim. Ayrıca destek ve yardımlarından dolayı bölüm hocalarıma, arkadaşlarıma, varlığıyla bana güç katan biricik oğlum Kerem Batur'a, eşime ve aileme çok teşekkür ederim.

Gül MUZAFFER

Trabzon 2016

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Anahtar Noktası Tabanlı Kopyala Yapıştır Sahteciliği Tespiti” başlıklı bu çalışmayı baştan sona kadar danışmanım Yrd. Doç. Dr. Güzin ULUTAŞ’ın sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 10/06/2016

Gül MUZAFFER

İÇİNDEKİLER

Sayfa No

Yüksek Lisans Tezi

ÖZET

ANAHTAR NOKTASI TABANLI KOPYALA YAPIŞTIR SAHTECİLİĞİ TESPİTİ

Gül MUZAFFER

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Yrd. Doç. Dr. Güzin ULUTAŞ
2016, 94 Sayfa

Günümüzde birçok alanda kullanılan sayısal görüntülerin doğruluğu ve güvenilirliği büyük önem arz etmektedir. Sayısal görüntüler üzerinde gerçekleştirilen pasif yöntemler arasında yer alan en yaygın sahtecilik türlerinden birisi de kopyala yapıştır sahteciliğidir. Kopyala yapıştır sahteciliğinin gerçekleştirilme kolaylığı görüntü doğrulama sistemlerinin bu tür sahteciliklerin tespitini etkin bir şekilde yapabilmesi gerekmektedir.

Tez çalışmasında, literatürde var olan anahtar noktası tabanlı kopyala yapıştır sahteciliği tespiti yöntemleri iyileştirilmiş olup ayrıca anahtar noktası tabanlı yeni bir tespit yöntemi önerilmiştir. Önerilen yöntemde nesne kapama amaçlı gerçekleştirilen kopyala yapıştır sahteciliğinde; literatürde var olan popüler yöntemlere kıyasla ön işleme gerek duymadan daha etkin tespit gerçekleştiren, ataklara karşı dayanıklı yeni bir yöntem önerilmiştir. Tez çalışması kapsamında önerilen yöntemlerin literatürde var olan yöntemlerle karşılaştırılması gerçekleştirilmiş ve deneysel sonuçlar rapor edilmiştir.

Anahtar Kelimeler: Sayısal Görüntü Güvenliği**,** Görüntü Doğrulama, Kopyala Yapıştır Sahteciliği, Anahtar Noktası Tabanlı Yöntemler, SIFT, SURF, ORB, AKAZE, RANSAC, LPQ, Gabor Filtresi, ROC Analizi

Master Thesis

SUMMARY

KEYPOINT BASED COPY MOVE FORGERY DETECTION METHODS

Gül MUZAFFER

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Assoc. Ass. Prof. Güzin ULUTAŞ
2016, 94 Pages

The accuracy and reliability of digital images are very important. Copy move forgery is one of the most common digital image forgeries. As such an action is easy to accomplish, the image authentication systems should carry out the detection of this kind of forgery in its best.

In this thesis study, current key point based forgery detection methods are analysed and improved. Moreover, a novel keypoint based copy move detection method is proposed. The proposed detection method is found to be effective in detecting copy move forgeries without requiring pre treatment which is a clear advantage as to compared to the literature. Our method is especially effective in hidden object forgeries. All of the results of the suggested methods are compared with the current methods in literature and experimental results are reported.

Keywords: Digital Image Security**,** Image Authentication, Copy Move Forgery, Keypoint Based Methods, SIFT, SURF, ORB, AKAZE, RANSAC, LPQ, Gabor Filter, ROC Analysis

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

SEMBOLLER DİZİNİ

SIFT	Ölçek Değişimsiz Özellik Dönüşümü (Scale Invarient Feature Transform)
SURF	Hızlandırılmış Dayanıklı Öznitelikler (Speed up Robust Feature)
ORB	Oriented FAST and Rotated BRIEF
BRIEF	Binary Robust Independent Elementary Features
FAST	Features from Accelerated Segment Test
AKAZE	Hızlandırılmış KAZE (Accelerated-KAZE)
LPQ	Yerel Faz Kuantalama (Local Phase Quantization)
PSF	Noktasal Yayılım Fonksiyonu (Point Spread Function)
RANSAC	Random Sample Consensus
RGB	Kırmızı, Yeşil, Mavi (Red, Green, Blue)
AWGN	Toplanır Beyaz Gauss Gürültüsü (Adaptive White Gaussian Noise)
SNR	Signal to Noise Ratio
ROC	Receiver Operating Characteristic
DPO	Doğru Pozitif Oran
YPO	Yanlış Pozitif Oran
AKD	Ayrık Kosinüs Dönüşümü(Discrete Cosine Transform)
ADD	Ayrık Dalgacık Dönüşümü(Discrete Wavelet Transfom)
TBA	Temel Bileşenler Analizi(Principle Component Analiys)
AFD	Ayrık Fourier Dönüşümü
SVD	Tekil Değer Ayrışımı(Singular Value Decomposition)
GPD	Gauss Piramit Dekombozisyonu (Gauss Pyramid Decomposition)
DoG	Gauss Uzay Farkı (Difference of Gaussian)
MLDD	Multi-Level Dense Descriptor
AOS	Additive Operator Splitting
FED	Fast Explicit Diffusion
LDB	Local Difference Binary
M-LDB	Modified Local Difference Binary
STFT	Short Time Fourier Transform

1. GENEL BİLGİLER

1.1. Giriş

İnternet kullanımının giderek artması ve görüntü yakalama cihazlarının maliyetindeki düşüşün sonucu olarak sayısal görüntülerin oluşturulması, erişilebilirliği ve iletiminde hızlı bir artış gözlemlenmektedir. Tıp, gazetecilik, hukuk gibi birçok alanda da sayısal görüntülerin kullanımı yaygınlaşmaktadır. Photoshop, GIMP, Corel Draw gibi görüntü düzenleme yazılımlıları ile birlikte sıradan bir insanın bile, tespiti zor olan görüntü sahteciliği yapabilmesi, kullanımı yaygınlaşan görüntülerin doğruluğunu sorgulanır hale getirmiştir.

Görüntü sahteciliğinin tarihi 1840'lı yıllarda Hippolyte adlı kişinin intihara teşebbüs ettiğini gösteren Şekil 1.1'de verilen sahte görüntünün üretilmesiyle başlamıştır. Bu kişi ayrıca 1839 yılında Daguerre ve Talbot tarafından ileri sürülen bir görüntü değiştirme yöntemini önermiştir [1].

Şekil 1. 1. Hippolyte Bayard'ın intihara teşebbüs ettiğini gösteren ilk sahte görüntü [1].

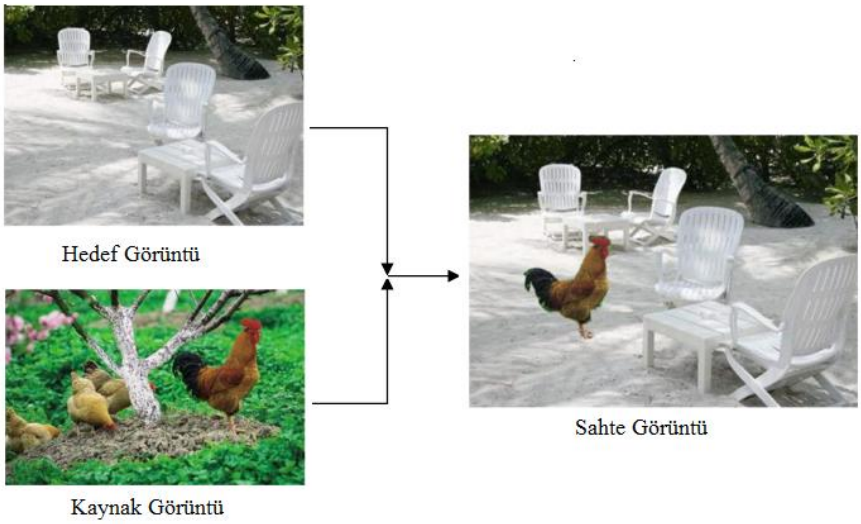
Sayısal görüntü güvenilirliğini kontrol etmek amacıyla literatürde birçok görüntü doğrulama yöntemi önerilmiştir. Genel olarak bu yöntemler aktif ve pasif doğrulama yöntemleri olmak üzere iki ana kategoride değerlendirilmektedir [2]. Aktif yöntemler

kendi içerisinde sayısal damgalama ve sayısal imzalama olmak üzere iki alt kategoriye ayrılmaktadır.

Sayısal damgalama özel oluşturulmuş damga bilgisinin görüntü içine gizlenmesi temeline dayanmaktadır. Çıkarılan damga bilgisinin kontrolünün yapılmasıyla görüntünün bir değişime uğrayıp uğramadığı hakkında bilgi edinilmektedir [3, 4]. Sayısal damgalama; telif hakkı koruma, kimlik tespiti, kopya engelleme, yayın denetleme ve verinin gerçekliğini kanıtlama gibi birçok alanda kullanılmaktadır. Damga bilgisinin görüntü oluşturulurken görüntünün içerisine yerleştirilmesi işleminin özel donanımlı kameralar veya sonradan yetkili yazılımlarca yapılmasına ihtiyaç duyulması, bu yöntemin dezavantajı olarak ortaya çıkmaktadır. Aktif yöntemlerin bir diğer kategorisi olan sayısal imzalar, görüntü transferinde görüntü ile birlikte görüntünün özelliklerinden elde edilen bir imza bilgisinin transfer edilmesi temeline dayanmaktadır. Bu yöntemde sayısal damgalamaya benzer şekilde, özel oluşturulmuş bir verinin iletimini gerektirmesi ve özel yazılımlara ihtiyaç duyması sebebi ile sayısal damgalamaya benzer dezavantaj içermektedir. Sonuç olarak görüntü doğrulama aşamasında her iki yöntem de ekstra bilgiye ihtiyaç duymaktadır. Ayrıca internet ortamında damga bilgisi içermeyen resimlerin varlığı, aktif yöntemlerin bu görüntülerin doğrulanması amacıyla kullanımını imkânsız hale getirmektedir.

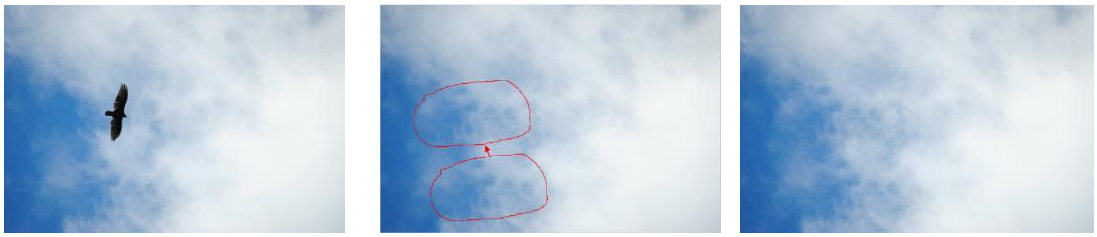
Pasif görüntü doğrulama yöntemlerinde, görüntünün doğrulanması işleminde görüntüden elde edilen istatistiksel özellikler kullanılarak görüntü doğrulaması gerçekleştirilmekte ve görüntü haricinde herhangi bir ek bilgiye ihtiyaç duyulmamaktadır [5]. İlgili yöntemlerin görüntü doğrulama işlemi için ek bir bilgiye ihtiyaç duymaması son zamanlarda araştırmacıların dikkatini bu yöne çekmiştir. Pasif görüntü doğrulama, görüntü birleştirme sahteciliği tespiti ve kopyala yapıştır sahteciliği tespiti olmak üzere iki alt kategoriye ayrılmaktadır.

Görüntü birleştirme sahteciliği, farklı görüntülerden elde edilen görüntü parçalarının birleştirilmesiyle yapılan sahtecilik türüdür [6]. Görüntü birleştirme sahteciliğine dair bir örnek de Şekil 1. 2’ de verilmiştir. Birleştirme ile oluşturulan sahte görüntünün bulanıklaştırma, JPEG sıkıştırma, gürültü ekleme gibi ek yöntemler kullanılarak fark edilemez hale getirilmesi sahtecilik tespitinin zorlu bir problem olmasına neden olmuştur.



Şekil 1. 2. Görüntü birleştirme sahteciliği örneği [7].

Pasif yöntemlerin tespit etmeye çalıştığı olan en popüler görüntü sahteciliği yöntemi, kopyala yapıştır sahteciliğidir [8]. Bu sahtecilik yönteminde görüntülenmesi istenmeyen nesnelerin gizlenmesi veya nesnelerin çoğaltılması amacıyla, belirli bir o bölgenin kopyalanarak aynı görüntüye yapıştırılmasıyla gerçekleştirilir. Şekil 1. 3' te kopyala yapıştır sahteciliğine dair bir örnek verilmiştir. Bu örnekte (a)' daki kuşun, (b)' de gösterildiği üzere, bulut bölgesinin kopyalanıp kuşun üstüne yapıştırılmasıyla bir sahte görüntü oluşturulmuştur.

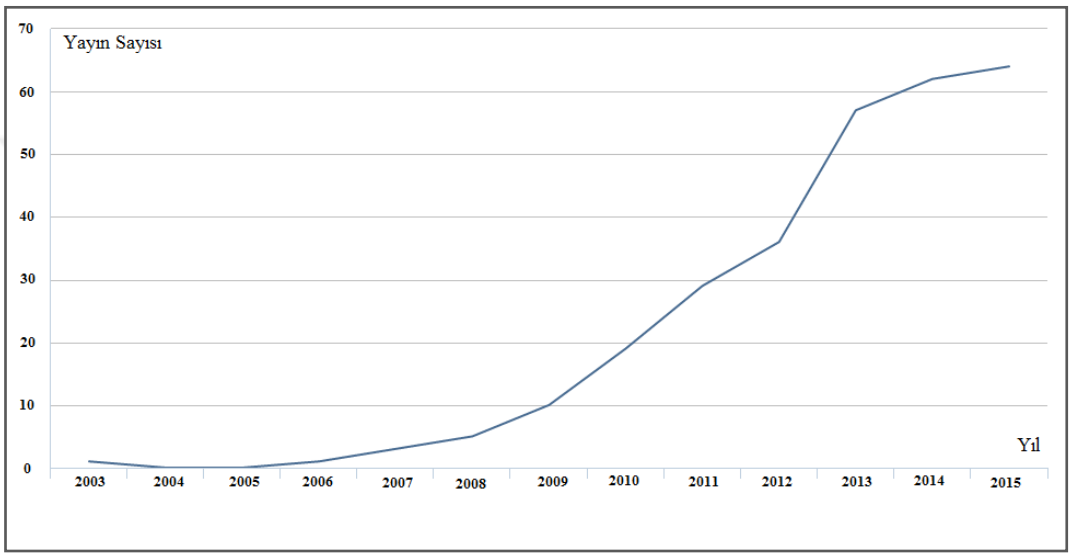


(a) (b) (c)

Şekil 1. 3. (a) Orijinal görüntü (b) Kopyala yapıştır sahteciliğinin gerçekleştirilmesi (c) Sahte görüntü

Kopyala yapıştır sahteciliği uygulanmış görüntünün yapısal analizi gerçekleştirildiğinde kopyalanıp yapıştırılan bölgeler arasında yüksek oranda benzerlik gözlemlenmektedir. Bu fikirden yola çıkarak kopyala yapıştır sahteciliğinin tespitine ilişkin ilk çalışma 2003 yılında gerçekleştirilmiştir [9]. Bu çalışmadan sonra kopyala yapıştır sahteciliği tespiti alanında birçok çalışma yapılmış ve bu alanda araştırmacıların

ilgisi artan bir şekilde devam etmektedir. Şekil 1. 4'te 2003-2015 yılları arasında Scopus'ta "copy move forgery" anahtar kelimesi ile elde edilen analiz raporlarına göre Scopus tarafından indekslenmiş yayınların yıllara göre dağılımı gösterilmiştir [10]. Bu bağlamda bu tez çalışmasında kopyala yapıştır sahteciliği tespitine ilişkin literatürde var olan yöntemler incelenmiş olup bu yöntemlerdeki tespit edilen problemlere karşı yeni yaklaşımlar kullanarak iyileştirmeler gerçekleştirilmiş ve ayrıca yeni bir anahtar noktası tabanlı kopyala yapıştır sahteciliği tespiti yöntemi önerilmiştir.



Şekil 1. 4. Kopyala yapıştır sahteciliği alanında Scopus tarafından indekslenmiş yayınların yıllara göre dağılımı [10].

1.2. Kopyala Yapıştır Sahteciliği Tespiti Yöntemleri

Kopyala yapıştır sahteciliği tespitinde kullanılan yöntemler genel çalışma yapısına göre blok tabanlı yöntemler ve anahtar noktası tabanlı yöntemler olmak üzere iki ana başlıkta değerlendirilmektedir. Blok tabanlı yöntemlerde test görüntüsü ön işlemden geçtikten sonra aynı boyutlu karesel veya dairesel bloklara ayrılmaktadır. Ayrılan bu bloklara ait özellik vektörleri, özellik çıkarma yöntemleri kullanılarak elde edilmekte ve özellik vektörlerinin eşleştirilmesinin ardından kopyalanan ve yapıştırılan bölgeler ortaya konulmaktadır. Yaygın olarak kullanılan bu çalışma yapısının ana adımları aşağıdaki gibi tanımlanmaktadır:

Adım 1: MxN boyutundaki sahte görüntü ilk olarak gri seviyeye dönüştürülür(Renk bilgisini kullanan algoritmalar haricinde)

Adım 2: Görüntü bxb boyutlu bloklara bölünür. Böylece $N_b = (M - b + 1)x(N - b + 1)$ adet blok elde edilmiş olur.

Adım 3: Her bloğa ait $1xK$ boyutlu f_i özellik vektörleri elde edilir. Daha sonradan kullanmak üzere sol üst koordinat bilgisi (x_i, y_i) de f_i vektöründe tutulur. Böylece f_i vektörünün boyutu $1x(K + 2)$ olur.

Adım 4: Bütün bloklara ait özellik vektörlerinin tutulduğu $N_b x(K + 2)$ boyutlu F özellik matrisi oluşturulur.

Adım 5: F özellik matrisinin satırları leksikografik olarak sıralanır. Böylece benzer vektörler birbirine yakınlaşır. Belirli bir eşik değerine göre en yakın komşuların arasından eşleşecek vektörler seçilir.

Adım 6: Eşleşen vektör çiftleri $i \neq j$ olmak üzere f_i ve f_j olsun. Eşleşen iki vektörün temsil ettiği bloklar arasındaki mutlak vektörel uzaklık s_i ile gösterilen kayma matrisinde tutulur. Eşitlik (1.1)' de kayma matrisinin ilgili satırına değer ataması verilmiştir. f_i^{K+1}, i ile gösterilen vektörün *(K+1).* Elemanıdır.

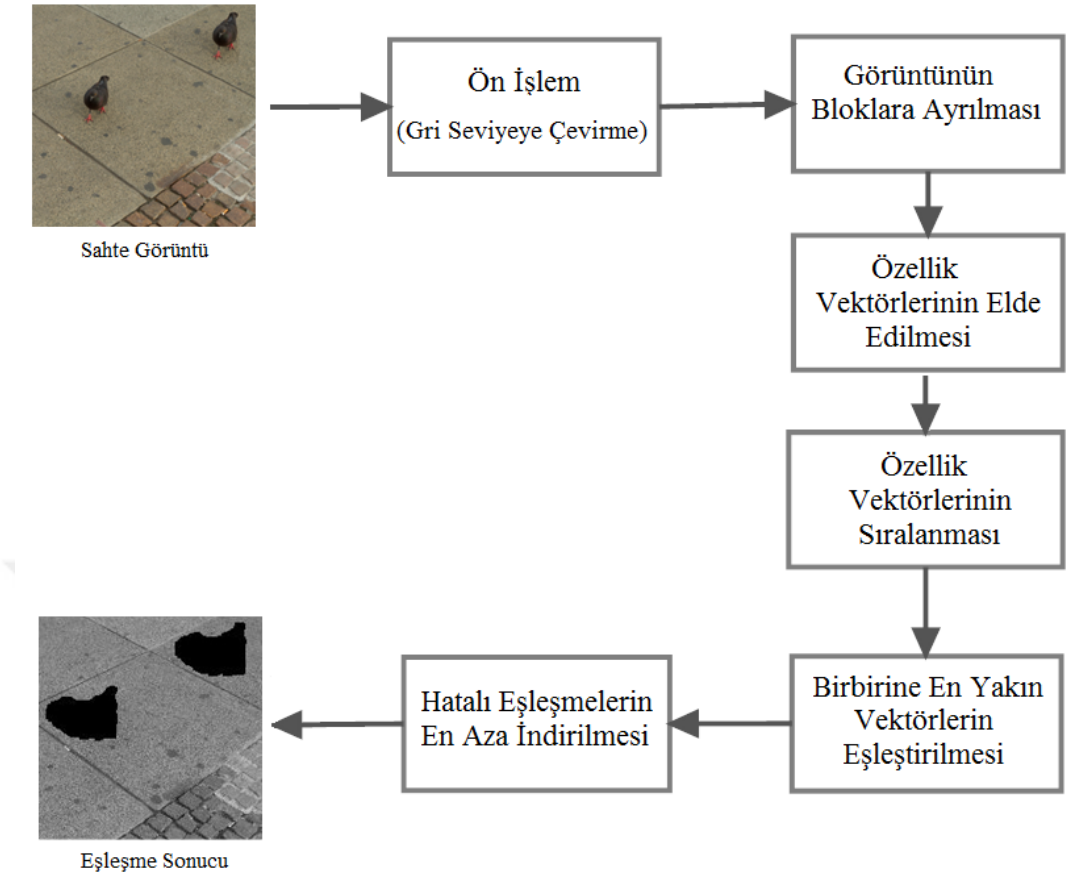
$$s_i(dx, dy) = (f_i^{K+1} - f_j^{K+1}, f_i^{K+2} - f_j^{K+2}) \tag{1.1}$$

Adım 7: Aynı kayma vektörünü oluşturan her eşleşen blok çifti için bir C sayacı oluşturulur.

$$C(dx, dy) = C(dx, dy) + 1 \tag{1.2}$$

Adım 8: Kopyalanıp yapıştırılan bölgeler için bloklar her zaman aynı değişim vektöründe bulundukları için belirli bir eşik değeri altında olan sayaç değerine sahip eşleşme vektörünü oluşturan blok çiftlerinin elenmesi işlemi gerçekleştirilir. Eşik değerini sağlayan blokların ise eşleşme işlemi gerçekleştirilerek vektörlerin temsil ettiği *BxB*'lik blokların boyanması gerçekleştirilir.

Adım 9: Son işlem adımında yanlış doğru olarak tespit edilen bölgeler morfolojik işlemler ile minimize edilir. Şekil 1. 5'de blok tabanlı kopyala yapıştır sahteciliğine ait blok diyagramı gösterilmiştir.



Şekil 1. 5. Blok tabanlı kopyala yapıştır sahteciliğinin tespiti blok diyagramı

Blok tabanlı yöntemlerde görüntüye ait bütün blokların özellik vektörleri çıkarıldığından hesaplama maliyeti yüksek olmaktadır. Araştırmacılar bu problemin üstesinden gelmek için anahtar noktası tabanlı yöntemleri önermişlerdir. Anahtar noktası tabanlı yöntemlerde görüntünün bütününden anahtar noktaları elde edilmektedir. Daha sonra anahtar noktalarının özelliklerini tanımlayan özellik vektörleri çıkartılmaktadır. Son olarak bu özellik vektörlerinden birbirine benzeyenlerin eşleştirmesi gerçekleştirilmektedir.

Anahtar noktası tabanlı kopyala yapıştır sahteciliği yöntemlerinin genel çalışma yapısının ana adımları aşağıdaki gibi tanımlanmaktadır:

Adım 1: MxN boyuta sahip sahte görüntü ilk olarak gri seviyeye dönüştürülür(Renk bilgisini kullanan algoritmalar haricinde)

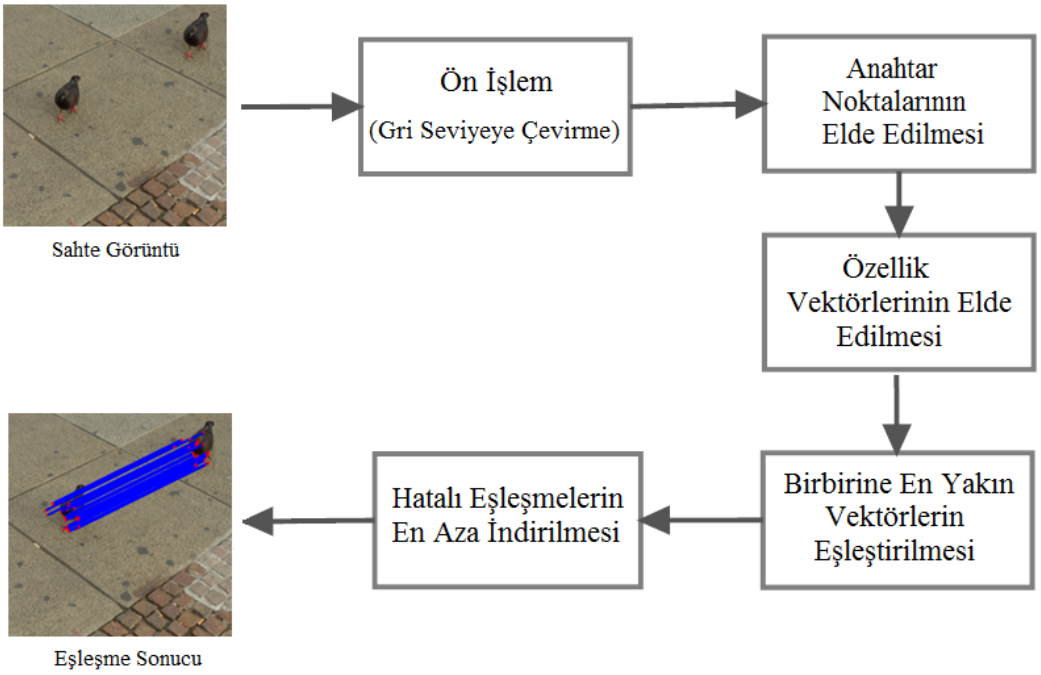
Adım 2: Anahtar noktası çıkarma algoritması ile birlikte tüm görüntüye ait anahtar noktaları elde edilir.

Adım 3: Anahtar noktalarına ait tanımlayıcı bilgileri tutan $1xK$ boyutlu özellik f_i tanımlama vektörleri elde edilir.

Adım 4: Daha sonradan kullanmak üzere anahtar noktalarının koordinat bilgisi (x_i, y_i) de f_i vektöründe tutulur. Böylece f_i vektörünün boyutu *1x(K+2)* olur.

Adım 5: Bütün anahtar noktalarına ait tanımlayıcı bilgiler tutulduğu N_b *x (K+2)* boyutlu *F* özellik matrisi oluşturulur.

Adım 6: Özellik vektörlerinin mutlak uzaklık değerinin hesaplanmasıyla birlikte birbirine en yakın anahtar noktaların eşleştirilmesi gerçekleştirilir. Şekil 1. 6'da anahtar tabanlı kopyala yapıştır sahteciliğine ait blok diyagramı gösterilmiştir.



Şekil 1. 6. Anahtar noktası tabanlı kopyala yapıştır sahteciliğinin tespiti blok diyagramı

1.3. Literatür Araştırması

Bu bölümde tezin kapsamında yer alan pasif doğrulama yöntemlerine ilişkin literatürde gerçekleştirilen yöntemler incelenecektir.

1.3.1. Blok Tabanlı Kopyala Yapıştır Sahteciliği Tespiti Yöntemleri

Kopyala yapıştır sahteciliğinin tespit edilmesine ilişkin literatürde yer alan ilk çalışma 2003 yılında Fridirich ve arkadaşları tarafından gerçekleştirilmiştir [9]. Bu

çalışmada görüntü 8x8 büyüklüğündeki bloklara ayrılmış ve oluşturulan her bir bloğa Ayrık Kosinüs Dönüşümü (AKD) uygulanması ile özellik vektörleri elde edilmiştir. Bloklardan üretilen özellik vektörlerinin leksikografik olarak sıralanmasının ardından, komşu vektörlerin birbirlerine olan benzerliğinin Öklid uzaklığı ile hesaplanmasıyla benzer özelliğe sahip bloklar belirlenmiş olur. Yapılan bu çalışma, sahte görüntünün JPEG sıkıştırmaya maruz kalması durumunda sonuç verirken, döndürme ve ölçekleme ataklarına karşı dayanıksızdır.

Popescu (2004) ve arkadaşları görüntünün ayrılan bloklarından [9]'da önerilen yöntem ile elde edilen özellik vektör boyutunun küçültülmesi ve doğrulama esnasındaki işlem karmaşıklığının azaltılması için Temel Bileşen Analizini (TBA) kullanmıştır [11]. Deneysel sonuçlar yöntemin Toplanır Beyaz Gauss Gürültüsü (AWGN), JPEG sıkıştırması ve bulanıklaştırma ataklarına karşı dayanıklı olduğunu göstermiştir.

Luo (2006) ve arkadaşları tarafından önerilen çalışmada bloklara ait özellik vektörleri üretilirken blokların yoğunluk bilgisinden faydalanılmıştır [12]. Görüntü bloklarının RGB (kırmızı, yeşil ve mavi) renk kanallarına ait ortalama yoğunlukları ve bazı yön bilgileri ile birlikte *1x7* boyutlu özellik vektörleri elde edilmiştir. Çalışmada geliştirilen yöntemin [9] ve [11]'deki çalışmalara kıyasla hesaplama karmaşıklığının daha düşük olduğu ve aynı zamanda gürültü, bulanıklaştırma ve bunların kombinasyonu şeklinde uygulanan ataklara karşı da daha dayanıklı olduğu bildirilmiştir.

Mahdian (2007) ve arkadaşları tarafından Bulanık momentler kullanılarak bulanıklaştırma ataklarına karşı dayanıklı bir yöntem önerilmiştir. Her bir bloktan *1x72* boyutlu özellik vektörleri elde edilmiştir [13]. Elde edilen özellik vektörlerinin boyut büyüklüğünden dolayı yazarlar eşleştirme zamanını iyileştirmek için özellik tanımlayıcı vektörlerin boyutlarını TBA kullanarak azaltmışlardır. Sonuçlarda [9, 11]'deki çalışmalara göre özellikle bulanıklaştırma atağı durumunda daha yüksek performansa sahip olduğu gösterilmiştir. Önerilen yöntemin ayrıca AWGN ve kayıplı JPEG sıkıştırma ataklarına karşı dayanıklı olduğu deneysel sonuçlarla birlikte ispatlanmıştır.

Myrna (2007) ve arkadaşları log-polar koordinat sistemi ve dalgacık dönüşümü temeline dayalı yeni bir kopyala yapıştır sahteciliği tespiti yöntemi önermişlerdir [14]. Görüntüye dalgacık dönüşümü uygulandıktan sonra bloklara ayırma işlemi gerçekleştirilmektedir. Ayrılan bloklar log polar koordinat sisteminde ifade edilmiş ve elde edilen vektörler leksikografik olarak sıralanmıştır. Sıralamanın ardından en yakın komşuların eşleştirilmesi gerçekleştirilmiştir. Önerilen bu yöntemde diğer yöntemlere göre,

dalgacık dönüşümünden dolayı daha az hesaplama karmaşıklığı içermektedir. Sahte görüntü üzerinde döndürme ve ölçekleme atakları olması durumunda bile önerilen algoritmanın etkin bir şekilde çalıştığı belirtilmiştir.

Kang (2008) ve arkadaşları kopyala yapıştır sahteciliği tespitinde Tekil Değer Ayrışımı (Singular Value Decomposition, SVD) yöntemini kullanmayı önermiştir [15]. Yöntem Gauss bulanıklaştırma, gürültü ekleme ve kayıplı JPEG sıkıştırmaları durumlarında bile kopyala yapıştır sahteciliğini gerçekleştirebilmektedir. Önerilen yöntemin [9], [11-13]'deki çalışmalara göre daha etkin ve daha düşük hesaplama karmaşıklığına sahip olduğu ve ayrıca gürültü ataklarına karşı daha dayanıklı olduğu belirtilmiştir.

Zhang (2008) ve arkadaşları görüntünün alt bantlarını elde etmek için ADD kullanmışlardır [16]. Faz korelâsyon bilgisi hesabı ile birlikte de blokların benzerlikleri test edilmiştir. Böylece diğer yöntemlere göre daha etkin ve ataklara karşı daha dayanıklı bir yöntem önerildiği belirtilmiştir. Ancak yöntemin en önemli dezavantajı ötelemeye duyarlı olmamasıdır.

Bayram (2009) ve arkadaşları tarafından önerilen bir diğer yöntemde, Fourier Mellin dönüşümü kullanılarak bloklardan özellik vektörü elde etmede kullanılmıştır [17]. Önerilen yöntemde ayrıca Counting Bloom filtresi kullanılarak vektör eşleştirme zamanı iyileştirilmiştir. Sonuçlarda yöntemin düşük derecede döndürme ataklarına karşı dayanıklı olduğu belirtilmiştir. Yöntem 10 derecenin üzerindeki döndürme ataklarında başarılı bir sonuç üretememektedir.

Wang (2009) ve arkadaşları Gauss Piramit Dekompozisyonu (GPD) ile boyut azaltımı gerçekleştirmiş ve görüntüyü karesel bloklar yerine dairesel bloklara ayrıştırmıştır [18]. Bu dairesel bloklara ait *1x4* boyutlu özellik vektörleri elde edilmiştir. Özellik vektörlerinin leksikografik olarak sıralanmasının ardından benzer blokların eşleştirilmesi gerçekleştirilmiştir. Yöntemin dönme, gürültü ekleme, bulanıklaştırma ve JPEG sıkıştırma ataklarına karşı dayanıklı olduğu da belirtilmiştir.

Bashar (2010) ve arkadaşları [11]'deki yaklaşımı geliştirerek dönme ve ters çevirme ile kopyala yapıştır sahteciliği uygulanan görüntüleri tespit etmek için ADD ve Çekirdek-TBA (ÇTBA) yöntemini kullanmışlardır [19]. Geliştirilen yöntemin [12]'deki TBA tabanlı yönteme göre daha etkili sonuç verdiği belirtilmiştir. Yöntemin ayrıca JPEG sıkıştırma ve gürültü ekleme ataklarına karşı da dayanıklı olduğu gösterilmiştir.

Khan (2010) ve arkadaşları [14]'te önerilen yöntemi optimize ederek daha düşük hesaplama karmaşıklığına sahip bir yöntem önermişlerdir [20]. Önerilen bu yöntem iki aşamadan oluşmaktadır. İlk aşamada ADD kullanılarak görüntü önce düşük boyutlu gösterime çevrilmiştir. Daha sonra bloklara ayrılan görüntüden elde edilen maksimum kontrasta sahip pikseller seçilip özellik matrisi oluşturulmuştur. Bu özellik matrisi sıralandıktan sonra satırlar arası faz korelâsyon hesabı yapılmış ve eşleşen bloklar yeni bir matriste tutulmuştur. İkinci aşamada ise farklı ADD seviyelerinde eşleşen blokların kontrolü yapılmıştır. Böylece daha dayanıklı bir kopyala yapıştır sahteciliği yöntemi önerilmiştir. Gürültü ve JPEG sıkıştırma ataklarına karşı dayanıklı olan bu yöntemin dönme ve ölçekleme ataklarına karşı ise dayanıklı olmaması en önemli dezavantaj olarak rapor edilmiştir.

Huang (2011) ve arkadaşları tarafından geliştirilmiş Ayrık Kosinüs Dönüşümü (AKD) tabanlı kopyala yapıştır sahteciliği yöntemi önerilmiştir [21]. Önerilen yöntemin çalışması şu şekildedir: Öncelikle görüntü bloklara ayrılır ve bu blokların özelliklerini çıkarmak için AKD uygulanır. Özellik vektörünün boyut azaltılmasını gerçekleştirmek için kırpma (truncating) işlemi gerçekleştirilir. Daha sonra leksikografik olarak sıralanan vektörlerin eşleştirilmesi gerçekleştirilmektedir. Sahte görüntü JPEG sıkıştırma, bulanıklaştırma ve AWGN gibi bozulmalara maruz kalsa bile önerilen yöntem sayesinde tespiti yapıldığı ortaya konmuştur.

Bravo-Solorio (2011) ve arkadaşları bloklara ait özellik vektörleri çıkararak benzerliklerin test edilmesi işlemi için Fourier dönüşümünün korelâsyon katsayılarını kullanmışlardır [22]. Önerilen yöntemde düşük entropiye sahip bloklar elenmiştir. Deneysel sonuçlar yöntemin yansıma, dönme, ölçekleme ve bunların kombinasyonu olan ataklara karşı dayanıklı olduğunu göstermiştir. Ayrıca yöntemin [14] ve [31]'deki çalışmalar ile kıyaslaması yapılıp üstünlüğü ortaya konulmuştur.

Gharibi (2011) ve arkadaşları görüntünün ayrılan bloklar üzerinde doku analizi yaparak, kopyala yapıştır sahteciliği tespiti alanında yeni bir yöntem önermişlerdir [23]. Blokların Gabor filtresi kullanılarak özellik vektörleri elde edilmiş olup bu özellik vektörlerinin eşleşme sürelerinin kısalması için TBA ile boyut azaltılması gerçekleştirilmiştir. Son olarak da özellik vektörlerinin sıralanması ve eşleştirilmesi işlemi gerçekleştirilmektedir. Görsel sonuçlar ile önerilen yöntemin literatürdeki benzer çalışmalara göre %10 daha fazla doğruluk oranına sahip olduğu ortaya konulmuştur.

Hsu (2012) ve arkadaşları görüntünün ayrılan bloklarından farklı ölçek faktörü, dönme açısı ve frekans bilgileri ile birlikte Gabor filtresi uygulamış ve bloklara ait Gabor özellik vektörleri elde etmiştir [24]. Önerilen yöntemin [23]'deki çalışma ile karşılaştırılması yapılmış ve doğrulama oranı açısından daha üstün performans sergilediği ortaya konmuştur. Deneysel sonuçlar yöntemin dönme ve ölçekleme ataklarına karşı dayanıklı olduğunu ve ayrıca dönme ve ölçekleme atağının birlikte uygulandığı görüntülerde de sahtecilik tespiti yapabildiğini göstermiştir.

Muhammad (2012) ve arkadaşları alt örneklemesiz Diyadik Dalgacık Dönüşümü kullanarak kopyala yapıştır sahteciliği tespiti yöntemi önermişlerdir [25]. Dalgacık alt bantları %50 örtüşen bloklara ayrılmaktadır. Blokların katsayılarının benzerlikleri kullanılarak eşleşme işlemi gerçekleştirilmektedir. Yöntemin JPEG sıkıştırma ve dönme ataklarına karşı dayanıklı olduğu belirtilmiştir.

Li (2013) ve arkadaşları dönmeden bağımsız yerel ikilik örüntü(rotation invariant Local Binary Patterns, LBP) tekniğini kullanarak, ayrılan bloklardan özellik vektörleri elde etmiş ve ardından eşleşme işlemi gerçekleştirmiştir [26]. Deneysel sonuçlarda önerilen yöntemin JPEG sıkıştırma, gürültü ve bulanıklaştırma gibi atakların yanı sıra dönmeye ve ters çevirme ataklarına karşı da dayanıklı olduğu gösterilmiştir.

Ryu (2013) ve arkadaşları tarafından, görüntünün karesel bloklarından dönmeden bağımsız özellik vektörleri elde etmek için Zernike Momentleri kullanılmıştır. Elde edilen özellik vektörlerini etkili bir şekilde eşleştirme amacıyla yerel duyarlı kıyım (Locality Sensitive Hashing, LSH) yönteminden faydalanılmıştır [27]. Hatalı eşleştirmeleri en aza indirmek için özellik uzayı hata azaltma yöntemi kullanmışlardır. Bu yaklaşım daha önceki uzaysal tabanlı yöntemlerde kullanılmamış olup daha doğru sonuçlar elde edilmesini sağlamıştır. Önerilen yöntemin JPEG sıkıştırma, bulanıklaştırma, beyaz Gauss gürültüsü ve ölçeklemeye karşı dayanıklı olduğu belirtilmiştir.

Lee (2015) ve arkadaşları tarafından yön histogramı (Histogram Of Gradient, HOG) bilgisi kullanılarak görüntünün bloklarına dair özellik vektörleri elde edilerek kopyala yapıştır sahteciliği tespiti yapılmıştır [28]. Elde edilen özellik vektörleri leksikografik olarak sıralanmış ve ardından benzer blok çiftlerin eşleştirilmesi gerçekleştirilmiştir. Çalışmanın aynı görüntüde çoklu kopyala yapıştır sahteciliği tespiti yapabildiği ortaya konulmuştur. Ayrıca küçük dereceli dönme, bulanıklaştırma, parlaklık değişimi ve renk azaltma gibi ataklara karşı da dayanıklı olduğu belirtilmiştir. Deneysel sonuçlar ile birlikte

önerilen yöntemin [9], [11] ve [35]'deki yöntemlerine göre daha etkin olduğu ispatlanmıştır.

Lee (2015) tarafından önerilen yöntemde ise blokların özellik vektörlerinin elde edilmesi öncelikle blokları temsil eden Gabor genlik değerleri hesaplanmaktadır [29]. Daha sonra yönlendirilmiş Gabor genlik değerlerine ait histogram bilgisi çıkarılarak istatistiksel özellikler elde edilmiştir. Elde edilen özellik vektörlerinin [13]'deki çalışmaya benzer şekilde eşleştirilmesi gerçekleştirilmiştir. Hatalı eşleşmelerin yok edilmesi içinde gürültü yok etme yöntemi önerilmiştir. Deneysel sonuçlar önerilen yöntemin hafif dönme, ölçekleme, JPEG sıkıştırma, bulanıklaştırma ve parlaklık değişimi ataklarına karşı dayanıklı olduğunu göstermiştir.

Bi (2016) ve arkadaşları tarafından, Çok Seviyeli Yoğunluk Tanımlayıcısı (Multi-Level Dense Descriptor, MLDD) çıkarma ve hiyerarşik özellik eşleştirme yöntemleri kopyala yapıştır sahteciliği tespitinde kullanılmıştır [30]. Önerilen bu çok seviyeli yoğunluk tanımlayıcısı, sırasıyla Renk Doku Tanımlayıcısı (Colour Texture Descriptor) ve Sabit Moment Tanımlayıcısı (Invariant Moment Descriptor) yöntemlerinin kullanımı ile elde edilmektedir. Görüntünün her pikselinin çok seviyeli yoğunluk tanımlayıcısı hesaplandıktan sonra hiyerarşik özellik eşleştirme yöntemi ile sırasıyla eşleşen bölgelerin tespiti yapılmaktadır. Aynı renk dokusuna sahip pikseller gruplandırılarak ayrık piksel grupları oluşturulmaktadır. Eşleştirme işlemi komşu piksel grupları ile birlikte yapılmaktadır. Önerilen bu yaklaşım ile birlikte hesaplama maliyeti düşürülmektedir. Hatalı eşleşmeler uyarlanabilir uzaklık ve yönlenim tabanlı filtreleme (Adaptive Distance and Orientation Based Filtering) ile minimize edilmiştir. Önerilen bu yöntemin geometrik bozulma, JPEG sıkıştırma, gürültü ekleme ve boyut azaltma gibi ataklara karşı dayanıklı olduğu gösterilmiştir.

1.3.2. Anahtar Noktası Tabanlı Kopyala Yapıştır Sahteciliği Tespiti Yöntemleri

Huang (2008) ve arkadaşları kopyala yapıştır sahteciliği tespitinde Ölçek Bağımsız Özellik Dönüşümü (Scale Invariant Feature Transform, SIFT) algoritmasını kullanmayı ilk olarak önermişlerdir [31]. Yöntemde öncelikle girdi görüntüsüne ait anahtar noktaları elde edilmekte ve ardından bu anahtar noktaların özellik vektörleri çıkartılarak eşleştirme gerçekleştirilmektedir. Deneysel sonuçlarda kopyala yapıştır sahteciliği uygulanan

görüntünün dönme, ölçekleme, gürültü ekleme, bulanıklaştırma ve JPEG sıkıştırma ataklarına maruz kalması durumunda bile başarılı olduğu ortaya konulmuştur.

Bu çalışmanın ardından Pan (2010) ve arkadaşları yeni bir SIFT tabanlı kopyala yapıştır yöntemi önermişlerdir [32]. Bu yöntemde SIFT ile birlikte elde edilen anahtar noktalarına ait özellik vektörlerinin de çıkarılması sonrasında birbirine en yakın komşu özellik vektörlerini bulmak için Best-Bin-First (BBF) algoritması uygulanmıştır. Korelâsyon haritalaması ile birlikte eşleşen bölgelerin sınırları belirlenmektedir Eşleşme sonrası ortaya çıkan bölgelerde uzaysal dönüşümü tutarlı olmayan eşleşmeler elimine edilerek hatalı eşleşmelerin önüne geçilmiştir. Bu yöntemin korelâsyon haritalaması ile eşleşen bölgenin sınırlarının belirlemesi sayesinde, eşleşme sonucunda sadece eşleşen anahtar noktalarının gösterildiği [31]'e göre daha etkin bir sonuç elde edilmiştir. Önerilen bu yöntemin ayrıca ölçekleme ve dönmeye karşı dayanıklı olduğu deneysel sonuçlar ile birlikte ispatlanmıştır.

Pan (2010) ve arkadaşlarının bir sonraki çalışmalarında ise SIFT algoritması ile elde edilen anahtar noktalarına ait çıkarılan özellik vektörleri BBF algoritması ile eşleştirildikten sonra hatalı eşleştirmeleri elimine etmek için Random Sample Consensus (RANSAC) algoritması kullanılmıştır [33]. Daha sonra korelâsyon haritası oluşturularak eşleşen bölgelerin sınırları belirlenmiştir. Kopyalanıp yapıştırılan bölgenin ölçekleme, döndürme, yansıma ve aydınlatma gibi ataklara maruz kaldığı durumlarda bile sahtecilik tespiti yapabildiği ispatlanmıştır. Ayrıca önerilen yöntemin [11] ve [31]'deki çalışmalar ile benzer performansa sahip olduğu ancak bu çalışmalarda kopyalanıp yapıştırılan bölgenin bozulmaya uğradığı durumda başarısız olduğu belirtilmiştir.

Xu (2010) ve arkadaşları Hızlandırılmış Dayanıklı Öznitelikler (Speed up Robust Feature, SURF) algoritmasını kullanarak test görüntüsünden anahtar noktalar çıkartarak bu anahtar noktalarına ait özellik vektörlerini elde etmişlerdir [34]. Önerilen bu yöntemde anahtar noktalarına ait özellik vektörleri iki gruba ayrılarak, gruplardaki birbirine en yakın vektörlerin tespiti yapılarak anahtar noktaların eşleştirilmesi gerçekleştirilmiştir. Yöntemin dönme, ölçekleme, gürültü ekleme, bulanıklaştırma gibi ataklara karşı dayanıklı olduğu deneysel sonuçlar ile birlikte ortaya konmuştur.

[31]'deki çalışmadan sonra Amerini (2011) ve arkadaşları SIFT algoritmasını hiyerarşik kümeleme ile birlikte kullanarak daha kapsamlı bir çalışma gerçekleştirmiştir [35]. Kopyalanan bölgenin çoklu yapıştırılması sonucu oluşan sahteciliğin tespiti de bu çalışma kapsamında değerlendirilmiştir. Çalışmada test görüntüsünden SIFT algoritması

ile elde edilen anahtar noktalarına ait özellik vektörleri hiyerarşik kümeleme ile eşleştirilmiştir. Hatalı eşleştirmeleri yok etmek için RANSAC algoritmasından faydalanılmıştır. Önerilen yöntemin JPEG sıkıştırma, gürültü ekleme, dönme ve ölçekleme gibi ataklara karşı dayanıklı olduğu gösterilmiştir.

Jaberi (2013) ve arkadaşları SIFT ile benzer özellikler içermesinin yanı sıra ayna yansımalarına karşı da dayanıklı olan MIFT (Mirror Reflection Invariant Feature) tabanlı kopyala yapıştır sahteciliği tespiti yöntemi önermişlerdir [36]. Önerilen yöntemin temel adımları; anahtar noktalarının çıkarılması, anahtar noktalarına ait afin dönüşümlerin tahmin edilmesi, yinelemeli olarak afin dönüşümün azaltılması, kopyalanıp yapıştırılan bölgelerin belirlenmesi şeklindedir. Anahtar noktaların eşleşmesinden sonra hatalı eşleşmelerin RANSAC ile elimine edilmesi gerçekleştirilmiştir. Önerilen yöntemin ölçekleme, dönme ataklarına dayanıklıdır ve ayrıca özellikle küçük bölgelerin kopyalanıp yapıştırılmasında yüksek doğruluk oranı elde edilmiştir.

Kiruthika (2014) ve arkadaşları SURF tabanlı çoklu kopyala yapıştır sahteciliği tespiti yöntemi önermiştir [37]. Gri seviyeye çevrilen görüntüden SURF algoritması ile anahtar noktaları ve bu onlara ait özellik vektörleri elde edilmiştir. Eşleşen anahtar noktalarının belirlenmesinde [35]'de tanımlanan genelleştirilmiş 2NN (generalized 2NN,g2NN) stratejisi olarak adlandırılan eşleşme algoritması kullanılmıştır. Daha sonra eşleşen noktalar üzerinde Kümelenmiş Hiyerarşik Kümeleme yaklaşımı kullanılarak hatalı eşleşme oranları azaltılmaya çalışılmıştır.

Zhu (2015) ve arkadaşları sahte görüntülerin tespiti için düşük hesaplama zamanı avantajı olan dönmeye karşı dayanıklı Oriented FAST and Rotated BRIEF (ORB) algoritmasını kullanmışlardır [38]. Yazarlar ORB algoritmasına ölçek bağımsız özellik kazandırmak için bu algoritmayı Gauss ölçek uzayında uygulamışlardır. Çalışmada ayrıca nesne kapama sahteciliği ele alınarak önerilen yöntemin başarısı vurgulanmıştır. Deneysel sonuçlarda yöntemin JPEG sıkıştırma, bulanıklaştırma, gürültü ekleme, dönme ve ölçek bağımsız olduğu gösterilmiştir. Çalışmanın, SURF tabanlı [34]'deki çalışma ve SIFT tabanlı [35]'deki çalışmalara göre daha etkin olduğu ispatlanmıştır.

Wenchang (2016) ve arkadaşları Parçacık Sürü Optimizasyonu (PSO) kullanarak SIFT tabanlı yeni bir anahtar noktası tabanlı kopyala yapıştır sahteciliği tespiti yöntemi önermişlerdir[39]. Yöntemin [34, 35]'deki çalışmalara göre daha etkin kopyala yapıştır sahteciliği tespiti yaptığı ortaya konulmuştur. Bu yöntemlerin az sayıda hatta hiç anahtar noktası bulmaması durumunda etkin kopyala yapıştır sahteciliği tespiti yapamayacağı

vurgulanmıştır. Çalışmada literatürdeki SIFT tabanlı kopyala yapıştır sahteciliği yöntemlerindeki klasik işlem adımları uygulandıktan sonra görüntünün özelliklerine göre her görüntü için otomatik olarak uygun parametre değerleri üretmektedir. Son adımda üretilen sonucun analizi yapılarak parametrelerden bir döngü içerisinde başa dönülerek N defa yeni grup üretilmektedir. Hatalı eşleşmeleri elemek için RANSAC ile önce eşleşen noktalar arasındaki uzaklığa bakarak da bir eleme yapılmaktadır. Önerilen yöntem ayrıca JPEG sıkıştırma, AWGN, dönme ve ölçekleme gibi ataklara karşı dayanıklıdır.

1.4. Anahtar Noktası Tabanlı Özellik Çıkarma Yöntemleri

Bu bölümde pasif doğrulama alanında yaygın kullanıma sahip olan SIFT, SURF ve ORB yöntemlerinin detaylarından bahsedilecektir. Aynı zamanda tez kapsamında nesne kapamaya dayalı sahteciliğin tespitinde kullanılan AKAZE yöntemi de ele alınacaktır.

1.4.1. Ölçekten Bağımsız Öznitelik Dönüşümü (Scale Invariant Feature Transform)

Ölçekten Bağımsız Öznitelik Dönüşümü (Scale Invariant Feature Transform, SIFT) algoritması, bir görüntüde ayırt edici yerel özellikleri çıkarmak için 1999 yılında David G. Lowe tarafından önerilmiştir [40]. Bu algoritma bir görüntüden ölçekleme, öteleme ve dönmeden bağımsız öznitelikler elde edilmesini sağlamaktadır. Ölçeksel uzaydaki ekstremum noktaların belirlenmesi, anahtar noktaların belirlenmesi, yön atama işlemi ve anahtar noktası özellik tanımlayıcılarının belirlenmesi SIFT algoritmasının temel adımlarıdır.

1.4.1.1. SIFT Anahtar Noktalarının Belirlenmesi ve Yön Atama İşlemi

$I(x,y)$ girdi görüntüsü için ilk olarak Eşitlik (1.3) kullanılarak farklı standart sapmalara sahip Gauss filtreleri $G(x,y,\sigma)$ ile konvolüsyonu sonucu elde edilen görüntüler ile Gauss ölçek uzayı oluşturulmaktadır.

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \tag{1.3}$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (1.4)$$

Eşitlik (1.4)'de verilen Gauss filtresinin hesaplanmasında birinci ve ikinci ölçek için sırasıyla σ_1 ve σ_2 arasındaki oran k kadardır. Lowe k değerini $\sqrt{2}$ olarak ele almıştır [41]. Bu durumda Gauss çekirdeği ölçek uzayı standart sapma değeri Eşitlik (1. 5)'deki gibi hesaplanmaktadır.

$$\sigma_n = k^{n-1}\sigma = \sqrt{2}^{\,n-1}\sigma \quad (1.5)$$

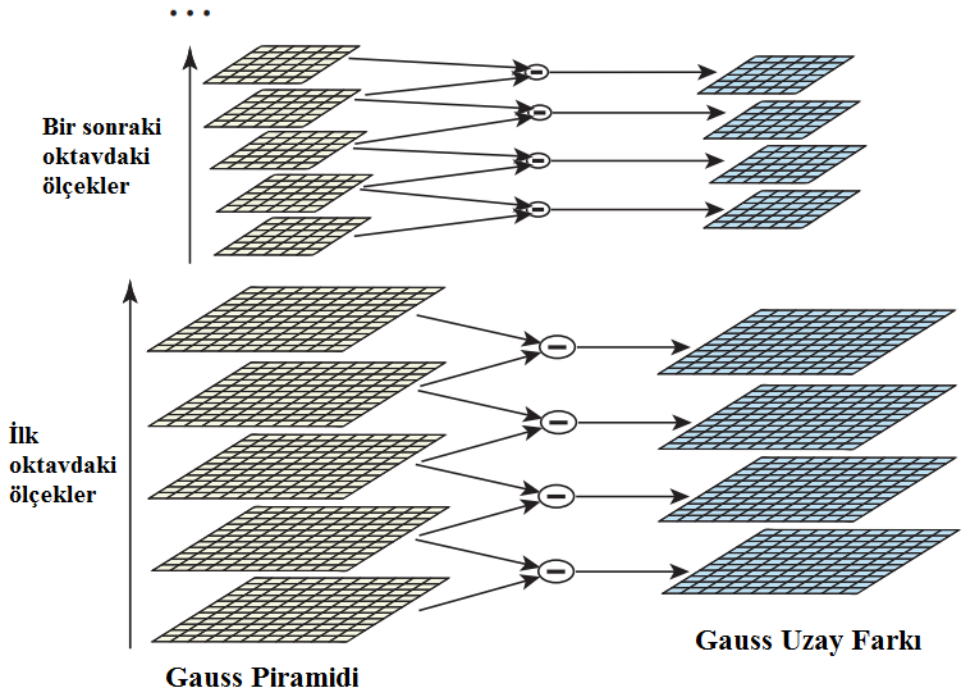
Ölçek uzayı, çarpım faktörü $k=2^{1/s}$ olacak şekilde (x,y, σ) parametrelerinden dolayı üç boyutlu bir karşılaştırmaya ihtiyaç duyulduğu için her biri *s+3* adet yumuşatılmış görüntü içeren oktav adı verilen serilere ayrılır. İkinci oktav, ilk oktavın σ_n kadar alt örneklenmiş ve boyutu yarı oranına indirgenmiş görüntü ile başlar. Sonraki her oktav için bu işlem bir önceki oktav kullanılarak oktav sayısı kadar benzer şekilde tekrarlanır.

Görüntünün ölçek uzayı belirlenmesi işleminden sonraki adımda ise her oktav için Gauss uzay farkı (Difference of Gaussian, DoG) hesaplanır. DoG uzayı, farklı standart sapmalara sahip Gauss filtreleri ile konvolüsyonu gerçekleştirilmiş görüntülerin farkları alınarak elde edilmektedir. Bu işlemin genel yapısı Şekil 1.7' de gösterilmiştir.

$$\begin{aligned} D(x, y, \sigma) &= \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (1.6)$$

Ekstremum noktalarının bulunması aşamasında ise, her bir DoG görüntüsünün alt, üst ve kendi ölçeğindeki her piksel çevresindeki sekiz piksel ve komşu ölçeklerde hizasındaki dokuzar piksel olmak üzere toplam *26* piksel ile karşılaştırılır. Eğer piksel o bölgenin yerel maksimumu veya yerel minimumu ise aday anahatar noktası olarak belirlenir.

DoG uzayı ile elde edilen aday özellik noktalarının elenmesi işlemi ise bir sonraki adımda gerçekleştirilmektedir. DoG operatörünün yoğun kenardan etkilenmesi ve gürültüye hassas olmasından dolayı, düşük çözünürlüğe sahip ve kenarlarda yer alan anahtar noktaları minimize edilmektedir.



Şekil 1. 7. İki ölçek uzay arasındaki farkların (DoG) bulunması [40].

Düşük çözünürlüğe sahip özellik noktalarının minimize edilmesi için DoG uzayında $D\ (x, y, \sigma)$ fonksiyonunun ikinci dereceden Taylor serisi kullanılmaktadır. Bu yaklaşım eşleşme ve durağanlık anlamında büyük bir gelişme sağlamaktadır. Eşitlik (1.7) ile özellik noktalarının yeni konumları belirlenmekte ve $D(\dot{x})$ elde edilmektedir.

$$\dot{x} = \frac{d^2D^{-2}}{dx^2}\frac{dD}{dx} \quad , \qquad D(\dot{x}) = D + \frac{1}{2}\frac{dD^T}{dx}\dot{x} \tag{1.7}$$

Bu fonksiyon her aday özellik noktası için bu hesaplanmakta ve $|D(\dot{x})|$ değerinin eşik değerinden küçük olması durumunda ilgili aday noktası elenmektedir. Yazarlar bu eşik değerini 0.5 olarak ele almıştır.

Kenar bölgesinde tespit edilen aday anahtar noktalarını elemek için (1.8)'de verilen Heissan matrisi kullanılmaktadır.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{1.8}$$

Heissan matrisi hesabı yapıldıktan sonra bu matrisin öz değerleri hesaplanmaktadır. Ancak matrisin öz değerinden çok, değersel oranından çıkarım yapılacağı için yalnızca oransal olarak ilgilenmek yeterli olacaktır.

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \ , \quad Det(H) = D_{xx}\, D_{yy} - \left(D_{xy}{}^2\right) = \alpha\, \beta \qquad (1.9)$$

(1.9)'daki determinantın negatif olması durumunda bükümün farklı işarete sahip olmasından ötürü bu nokta elenir.

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\ \beta} = \frac{(r\beta+\beta)^2}{r\ \beta^2} = \frac{(r+1)^2}{r} \ , \quad \frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \qquad (1.10)$$

Eşitlik (1.10) yardımıyla elde edilen oran eşik değerinden küçük ise aday özellik noktası elenmektedir. Yazarlar *r=10* değerini kullanmışlardır. Böylece kesin anahtar noktaları belirlenmiş olur.

Anahtar noktalarına dönmeden bağımsızlık kazandırmak için bir sonraki adımda ise belirlenen anahtar noktalarına yön ataması yapılmaktadır. Her özellik noktası etrafında gradyan yönleri ve büyüklükleri hesaplanmakta ve bu bölgedeki en belirgin yön bulunmaktadır. Bulunan en belirgin yön özellik noktasının yönü olarak atanmaktadır.

Her bir $L(x\,, y)$ görüntü örneği için, özellik noktası etrafındaki her piksel için, pikseller arasındaki farklılıktan yola çıkılarak (1.11) kullanılarak ilgili nokta için gradyan büyüklüğü $m(x, y)$ ve yönelimi $\theta(x, y)$ hesaplanmaktadır.

$$m(x, y) = \sqrt{(L(x+1\,, y) - L(x-1, y)^2 + ((L(x, y+1) - L(x, y-1))^2}$$
$$\theta(x, y) = tan^{-1} \frac{L(x,y+1)-L(x,y-1)}{L(x+1\,,y)-L(x-1,y)} \qquad (1.11)$$

Her biri diğerinden *10* derece açı farkına sahip olacak şekilde *360* derecelik yön aralığını kapsayan *36* adet bin'den oluşan bir yön histogramı oluşturulur. Bu histogram anahtar noktasının ölçeği olan σ 'nın *1,5* katı kadar genişlikteki Gauss aralıklı dairesel penceresindeki özellik noktalarının gradyan büyüklük değeri olan $m(x, y)$ eklenmesi ile elde edilir. Örneğin pencere içerisindeki pikselin yönelimine en yakın binin değerine o pikselin gradyan büyüklüğü eklenir. Bu işlem pencere içerisindeki tüm pikseller için uygulanır.

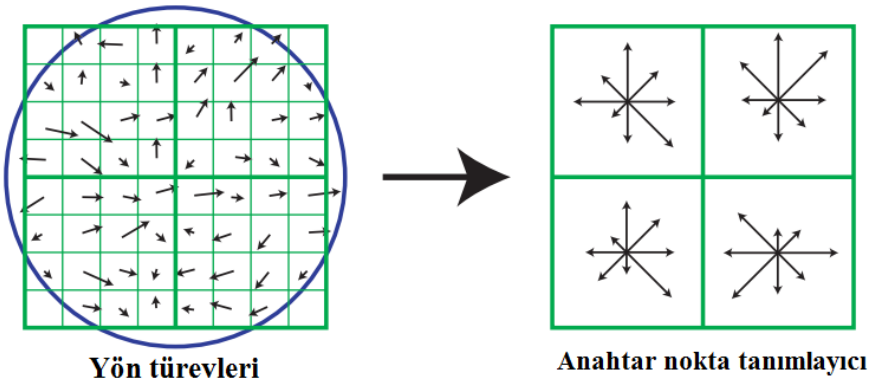
Oluşturulan histogramdaki en yüksek tepe noktasına sahip bin yönelim açısı baskın değerini verir. Ayrıca bu değerin *%80*'inden büyük olan diğer tepe noktalarına sahip yönelimler kullanılarak aynı noktada fakat farklı yönelime sahip anahtar noktalar oluşturulur. Noktaların sadece yaklaşık *%15*'inin sahip olabildiği bu durum yönelimin kararsızlığına sebep oluyormuş gibi gözükse bile anahtar noktasına ek özellik katarak noktanın benzersiz bir yapıya kavuşmasını sağlar. Bu da eşleşmenin kararlılığına katkı sağlar.

1.4.1.2. SIFT Özellik Tanımlayıcılarının Elde Edilmesi

Bu adımda daha önceki adımlarda konum, ölçek ve yön bilgisi atanan anahtar noktalarının birbirinden ayırt edilebilmesi için her özellik noktasına özel tanımlayıcılar oluşturulmaktadır. İlk aşamada, her özellik noktası etrafında *16x16'*lık blok alınmakta ve birbirleri arasında *45* derece açı farkı olacak şekilde sekiz yönelime sahip yönelim histogramı içeren *4x4* lük *16* bloğa bölünerek örnekleme alanı oluşturulmaktadır. Her okun yönelimi histogramın yönelim bilgisini; büyüklüğü ise bir hesaplanan gradyan büyüklüğünü ifade eder.

Bir histogramdan diğerine ya da bir yönden diğerine yumuşak örnek kaydırmalar için yani değişimlerdeki sınır etkisini azaltmak için diğer bir ifadeyle sekiz yönelime sahip histograma olan etkisini düzgün şekilde dağıtmak için, üçlü-doğrusal (tri-linear) interpolasyon kullanılarak her bir gradyan örnek değeri komşu histogram binlerine dağıtılmıştır. Yani bindeki her bir giriş, her bir boyut için *(1-d)* ağırlığıyla çarpılmıştır. Burada d, bin'in merkez değerinden uzaklığını vermektedir.

Şekil 1.8'de *8x8* örnek kümesinden hesaplanan *2x2* boyutlu tanımlayıcı dizisi gösterilmektedir. Tanımlayıcı Şekil 1.8'de görülen okların uzunluğuna karşılık gelen tüm yön histogram girişlerinin değerlerini içeren bir vektör biçimindedir. Fakat en iyi sonuçların her birinde sekiz yön bini olan *4x4* boyutlu histogram dizisiyle gerçekleştirilebileceği gösterilmiştir. Bu durumda *4x4x8=128* elemanlı özellik vektörü elde edilmiş olunur [40].



Şekil 1. 8. Görüntü gradyanı ve anahtar nokta tanımlayıcılar [40].

1.4.2. Hızlandırılmış Dayanıklı Öznitelikler (Speed up Robust Feature)

Hızlandırılmış Dayanıklı Öznitelikler (Speed up Robust Feature, SURF) algoritması bir görüntüde döndürme, ölçekleme ve ötelemeden bağımsız olarak yerel özellik noktalarının belirlenmesi için ilk olarak 2006 yılında Herbert Bay tarafından geliştirilen özellik çıkarma algoritmasıdır [41]. Bu yöntem, Lowe tarafından önerilen SIFT [40] yöntemi ile benzer adımlara sahip olmasının yanı sıra önemli iyileştirmelere de sahiptir. SURF algoritması herhangi bir performans kaybı olmadan SIFT'e göre anahtar noktalarını daha hızlı tespit edebilmektedir.

1.4.2.1. SURF Anahtar Noktaları Bulma

SURF algoritmasının temeli tümlev görüntülere ve Hessian matrisi ile birleştirilmiş konvolüsyon işlemine bağlıdır. (1.12)'de verilen Hessian matrisinin farklı görüntü bölgesi ortaya çıkarma özelliğinden yararlanarak görüntüdeki anahtar noktalar bulunur. Tümlev görüntü yaklaşımı ise hesaplama süresini oldukça düşürmektedir. Hessian matris determinantı ölçüt olarak kullanılarak bölgeler arasındaki değişimler hakkında bilgi edinilmektedir.

$$H = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{1.12}$$

Burada L_{xx} (1.13)'de görüldüğü gibi ikinci derece türevin konvolüsyon sonucudur ve L_{xy} ve L_{yy} 'de benzer şekilde elde edilir.

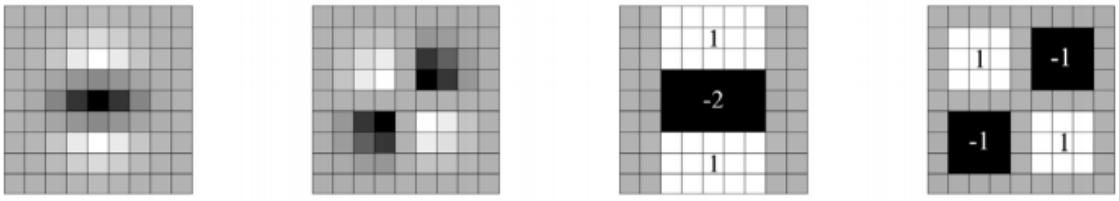
$$L_{xx}(x,\sigma) = I(x) * \frac{d^2}{dx^2} g(\sigma) \tag{1.13}$$

Konvolüsyon işlemleri hem hesaplama karmaşıklığını düşürmekte hem de tümlev görüntülerin hesaplanmasını kolaylaştırmaktadır. Bu şekilde daha hızlı sonuç elde edilmektedir.

Tümlev görüntünün elde edilmesi işleminde ise Eşitlik (1.14)'deki gibi görüntünün her pikseli için bu değer hesaplanarak yeni bir görüntü elde edilmektedir. Bu görüntü *x* ve *y* etrafındaki pikseller arasında kalan dikdörtgen bölgesinin piksel değerlerinin toplamından oluşmaktadır.

$$I\sum(x,y) = \sum_{i=0}^{i\leq x} \sum_{j=0}^{j\leq y} I(i,j) \tag{1.14}$$

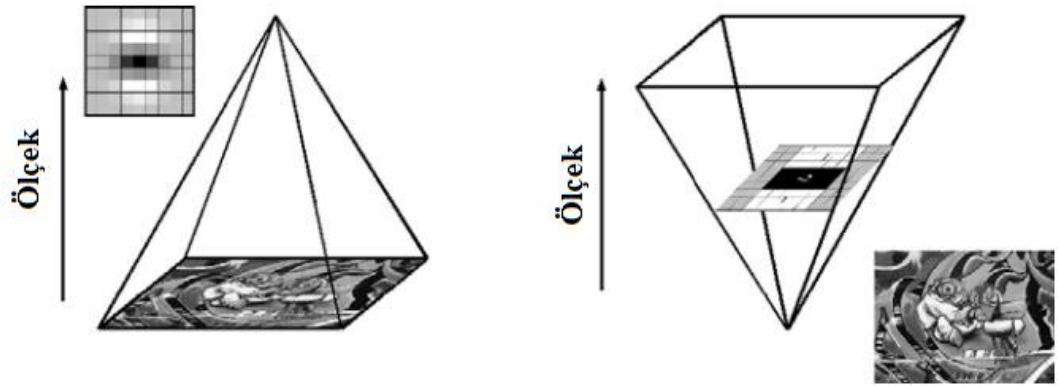
Hessian matris için kullanılan Gauss filtresi, *G(σ)* uygulanmadan önce ayrıklaştırılıp kırpılması gerekmektedir. SURF algoritması bu süzgeçleri kutu süzgeçlerle birlikte kullanmaktadır. Şekil 1.9'da *9x9* boyutunda *σ = 1.2* değerli Gauss filtresi kullanılarak elde edilmiş ölçek uzayın en alt seviyesi bulunmaktadır. Burada gri renkli alanlar sıfırı, beyazlar pozitifi ve siyahlar ise negatifi temsil etmektedir.



Şekil 1. 9. Soldan sağa doğru: y yönünde ve xy-yönüne 2. Derecede Gaussian türevleri ve yazarların bu türevlerin kutu filtreleri ile önerdikleri yaklaşımı [41].

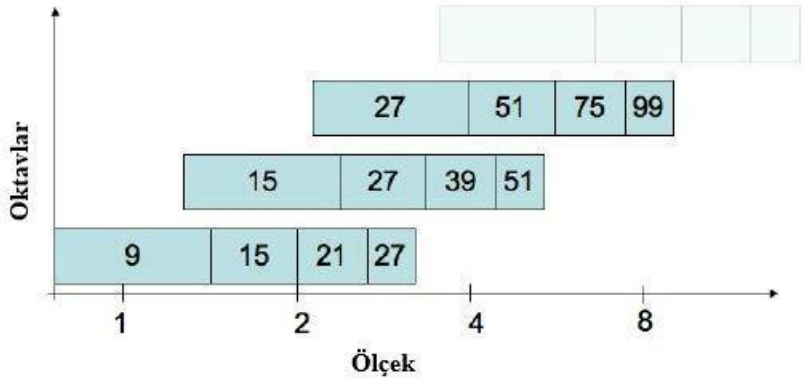
Kutu süzgeçleri ve tümlev görüntülerin kullanımından dolayı SIFT'de olduğu gibi bir önceki filtrelenmiş katmandaki görüntüye aynı filtre uygulanamamaktadır. Bu yüzden değişik boyutlarda ve ölçeklerde kutu süzgeçleri tümlev görüntülere uygulanarak ölçek-uzay yapısı oluşturulmaktadır. Bu yüzden ölçek uzayı tekrarlamalı bir şekilde boyutu

azalan görüntülerin yerine artan ölçekli görüntülerden oluşmaktadır. SURF algoritmasında oluşturulan piramitsel ölçek uzayı Şekil 1. 10'da sol taraftaki gibidir.



Şekil 1.10. SURF algoritmasında oluşturulan piramitsel ölçek uzayı [41].

Ölçek uzayının oluşturulması işlemine 9x9'luk filtre ile başlanmakta ve daha sonra filtre boyutu sırasıyla 15x15, 21x21 ve 27x27 olacak şekilde uygulanmaktadır. Şekil 1.11'de kutu filtreleri ile oluşturulan piramidin filtre boyutlarının grafiksel gösterimi verilmiştir.



Şekil 1.11. Üç oktava sahip piramitsel yapıyı oluşturmak için kutu filtrelerinin boyutları [41].

Oluşturulan ölçek-uzay yapısında Hessian determinantlarının sonuçlarına göre özellik noktaları çıkarılmaktadır. Ardışık üç ölçekten *3x3*' lük alanlar seçilerek toplamda *3x3x3=27* tane piksel arasında en yüksek gradyan değerine sahip piksel özellik noktası olarak belirlenmektedir.

1.4.2.2. SURF Özellik Tanımlayıcısı Çıkarma

Özellik noktalarına tanımlayıcı atama işleminde ilk adım olarak anahtar noktası merkez olacak şekilde karesel alanlar oluşturulur. Özellik noktasının bunduğu ölçek *s* olarak alındığında bu karesel alanların büyüklüğü *20s* olacak şekilde alınmalıdır. Belirlenen bu alan daha sonra büyüklüğü *5s* olacak şekilde *4x4'*lük karelere bölünür. Bu *4x4'*lük alanlara Haar dalgacık filtresi yatay ve dikey şekilde uygulanarak *x* ve *y* yönündeki türevler hesaplanmaktadır ve sırasıyla *dx* ve *dy* elde edilmektedir (filtre boyutu *2s*'dir). Geometrik bozulmalara ve konum hatalarına karşı dayanıklılığı artırmak için anahtar noktası merkez olacak şekilde *dx* ve *dy* yanıtları öncelikle $\sigma = 3.3\,s$ parametresi ile Gauss ağırlıklandırılması yapılır. Daha sonra her alt alan için elde edilen *dx* ve *dy* değerleri toplanır ve bu toplamlar tanımlayıcı vektörün ilk kısmını oluştururlar. Ayrıca tanımlayıcının kutupsal yoğunluk değişimleri hakkında bilgi de tutması için bu yanıtların mutlak değerlerinin ($|dx|$ ve $|dy|$) toplamları da elde edilir. Böylece her alt bölge dört boyutlu tanımlayıcı vektöre sahip olmuş olur. v= ($\sum dx, \sum dy, \sum|dx|, \sum|dy|$). Her *4x4* boyutlu alt vektör için bu dört boyutlu vektör çıkartılır. Dolayısıyla *4x(4x4)=64* boyutlu tanımlayıcı vektör oluşturulmuş olur.

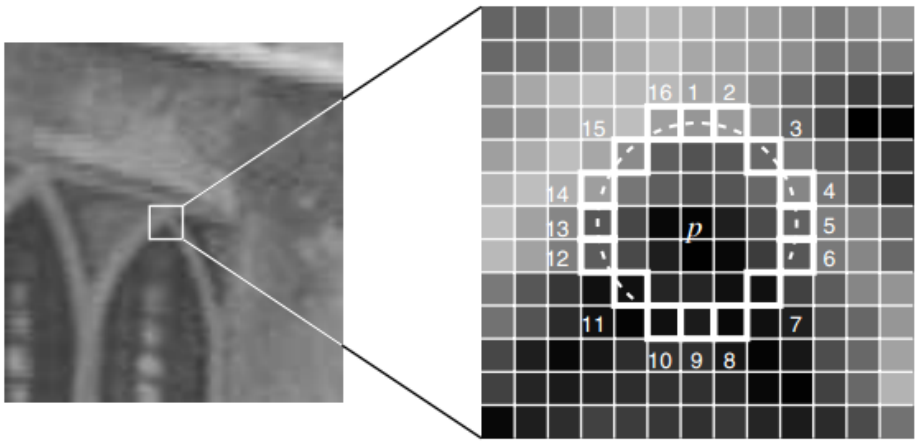
1.4.3. ORB (Oriented FAST and Rotated BRIEF)

Rublee ve arkadaşları 2011 yılında, SIFT ve SURF algoritmalarına alternatif olarak bir yöntem önermişlerdir [42]. Önerilen bu yöntem iyi performans ve düşük maliyet özelliklerine sahip olan FAST (Features from Accelerated Segment Test) ve BRIEF (Binary Robust Independent Elementary Features) algoritmaları üzerine kuruludur.

SIFT, SURF gibi anahtar noktası çıkarma yöntemleri yönelim operatörüne sahip iken FAST algoritmasında böyle bir özellik yoktur. Bu yüzden dönmeye karşı duyarlılık kazandırmak için ORB algoritmasında, FAST algoritmasıyla bulunan özellik noktalarına ait yönelim bilgisi elde edilip BRIEF sonucu çıkarılan tanımlayıcıyı yönlendirmiştir. ORB algoritmasının temel adımları: FAST ile anahtar noktasının tespit edilmesi, anahtar noktalarının yönelimlerinin bulunması, yönlendirilmiş BRIEF tanımlayıcılarının oluşturulması şeklindedir.

1.4.3.1. FAST ile Anahtar Noktalarının Elde Edilmesi

FAST (Features from Accelerated Segment Test) algoritması Edward Rosten ve Tom Drummond tarafından hızlı köşe noktası çıkarma yöntemi olarak önerilmiştir [43,44] . Bu yöntemde bir aday noktanın köşe olup olmadığına karar vermek için *16* piksellik Bresenham çemberi kullanır.



Şekil 1.12. FAST köşe bulma algoritmasında Bresenham çemberi çizimi [44].

Şekil 1.12'de görüldüğü gibi parlaklık değeri I_p olan bir *p* aday noktası seçildiğinde saat yönünde her bir piksel 1'den 16'ya kadar numaralandırılır. p noktasının köşe noktası olup olmadığına karar vermek için aşağıdaki durumlar kontrol edilir.

1. Durum*:* I_p değerini çemberdeki *1, 5, 9* ve *13* değerleri olan dört pusula yönündeki pikseller ile karşılaştırılması işlemi gerçekleştirilir. *t* eşik değeri olmak üzere, eğer seçilen dört nokta arasından en az üç noktanın piksel değeri, $I_p + t$ değerinden büyük veya $I_p - t$ küçük değil ise, *p* noktası köşe nokta olarak seçilir.

2. Durum*:* 1. Durumum tersi durumunda ise yani eğer en az üç noktanın piksel değeri $I_p + t$ değerinden büyük veya $I_p - t$'den küçük ise, çember üzerinde yer alan 16 pikselin hepsine bakılır. *16* piksel arasından en az *12* bitişik noktanın verilen şartı sağlaması gerekir. Bu işlem, görüntüdeki geriye kalan tüm noktalar için uygulanır.

ORB algoritmasında iyi performansından ötürü dairesel çemberin yarıçapı 3 ve karşılaştırmada kullanılan *t* eşik değeri 9 (FAST-9) olarak alındığı belirtilmiştir [42].

ORB ayrıca değişen ölçeklerde değişmeyen anahtar noktaları bulmak için görüntünün ölçek piramidini çıkararak FAST yöntemi uygulanarak çıkartılmaktadır. Her

bir ölçekte, anahtar noktalar FAST yöntemini uygulayarak çıkartılır. Anahtar noktalar elde edildiğinde, Harris köşe metriği kullanılarak anahtar noktalar sıralanır ve sadece en üsten N tane nokta belirli bir eşik değerine bağlı olarak seçilir.

1.4.3.2. Yönelimlerin Bulunması

ORB, kolay ve etkili köşe yöneliminin bulunmasını sağlayan yoğun ağırlık merkezi temeline dayanır. Yoğun ağırlık merkezi köşelerin yoğunluklarının merkezleri dengeledikleri varsayılmaktadır. Eşitlik (1.15) ile elde edilen köşe noktasının moment değerleri yardımıyla merkez koordinatları elde edilir.

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad , C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{1.15}$$

Merkez (C) ve köşe noktası (O) arasında $\overrightarrow{OC}$ vektörü olduğu düşünülürse bu aradaki açı değeri ise aşağıda verilen Eşitlik (1.16) ile bulunur.

$$\theta = atan\left(\frac{m_{01}}{m_{00}} / \frac{m_{10}}{m_{00}}\right) = atan\left(\frac{m_{01}}{m_{10}}\right) \tag{1.16}$$

1.4.3.3. BRIF ile Tanımlayıcıların Elde Edilmesi

BRIEF algoritması özellik noktalarına ait ikili bir tanımlayıcı üreten algoritmadır [46]. BRIEF yönteminde bir parça (patch) alınır ve gürültüye karşı bir önişlem olarak yumuşatılır. Bu yumuşatma işlemi tümlev görüntülerin alınmasıyla gerçekleştirilmiştir. Yumuşatılmış *31x31*'lik bir bölge olarak p ele alındığında ilgili bölge Eşitlik (1.17)'de gösterilen ikilik teste tabii tutulur.

$$\tau(p; x, y) \coloneqq \begin{cases} 1: \ p(x) < p(y) \\ 0: \ p(x) \geq \ p(y) \end{cases} \tag{1.17}$$

Burada $p(x)$, p parçasında x noktasının yoğunludur. Özellik vektörü olan $f_n(p)$ (1.18)' deki gibi n adet ikilik test yapılarak oluşturulur.

$$f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tag{1.18}$$

[46]'da yumuşatma işlemi için birçok dağılım yöntemi gözlemlenmiş olup ORB'de Gauss dağılımının en iyi performansa sahip olmasından ötürü seçildiği belirtilmiştir. Bu çalışmada aynı zamanda vektör uzunluğu *n=256* olarak alınmıştır [43].

ORB'de rotasyona karşı dayanıklılık kazandırmak için anahtar noktalarının yönelim bilgisi ile BRIEF sonuçları ilişkilendirilerek yönlendirilmiş BRIEF özellik tanımlayıcıları elde edilmektedir.

Herhangi x_i ve y_i noktası için elde edilen n tane ikilik test sonuçları için $P = \begin{bmatrix} x_1 \dots x_n \\ y_1 \dots y_n \end{bmatrix}$ matrisi tanımlandığında θ_i yönlenim açısı için dönme matrisi $R_{\theta_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}$ kullanarak yönlendirilmiş matris $P_\theta = R_\theta . P$ şeklinde elde edilir. Bu şekilde ORB tanımlayıcısı P_θ matrisini kullanarak Eşitlik (1.19)'daki gibi hesaplanır.

$$ORB(i) = f_n(p) | (x_i , y_i) \in P_\theta \tag{1.19}$$

1.4.4. AKAZE-Hızlandırılmış KAZE

Doğrusal olmayan ölçek uzayını kullanan anahtar noktası çıkarma algoritmalarından olan AKAZE (Accelerated-KAZE) algoritması Pablo ve arkadaşları tarafından 2013'te önerilmiştir [50]. Bu algoritma yine aynı yazarların önceki çalışması olan KAZE Özellik tanımlayıcısı algoritmasının geliştirilmesiyle elde edilmiştir [49].

Klasik anahtar çıkarma yöntemlerinden olan SIFT [40] ve SURF [41] algoritmaları çoklu ölçek yapısını kullanan algoritmalardır. Bu iki yaklaşım da Gauss ölçek uzayını kullanarak özellik tanımlamasını gerçekleştirmektedir. SIFT Gauss ölçek uzayını piramitsel yapıyı oluşturmak için kullanırken, SURF algoritması kutu filtreleri aracılığıyla yaklaşık Gauss türevlerini kullanırlar. Bu metotların en büyük dezavantajı nesne sınırlarının koruyamamasıdır. Çünkü Gauss bulanıklaştırması detayları ve gürültüyü aynı derecede bulanıklaştırmaktadır. Alcantarilla ve arkadaşları bu problemin üstesinden gelebilmek için 2012 yılında KAZE özellik çıkarma algoritmasını önermişlerdir [49]. Bu yöntem kullandıkları doğrusal olmayan filtreleme sayesinde SIFT ve SURF'e göre tekrar edilebilirliği ve ayırt edilebilirliği artırmıştır. KAZE doğrusal olmayan difüzyon denklemlerini çözmek için Additive Operator Splitting (AOS) yaklaşımını kullanmaktadır.

Ancak AOS yüksek hesaplama karmaşıklığı problemini doğurmuştur. KAZE yönteminin hesaplama karmaşıklığı, bu yöntemin esas dezavantajı olarak ortaya çıkmaktadır.

KAZE özellik çıkarma yönteminin kullandığı doğrusal olmayan difüzyon filtrelemenin sağladığı avantajı koruyarak daha hızlı bir şekilde özellik çıkarmak için aynı yazarlar tarafından doğrusal olmayan ölçek yapısını oluşturmak için Fast Explicit Diffusion (FED) matematiksel yapısı kullanılmıştır. Ayrıca FED'in KAZE özellik tanımlayıcı algoritmasında kullanılan AOS şemasına göre daha kolay uygulanabildiği ve daha doğru olduğu belirtilmiştir. FED yapısının önemli ölçüde sağladığı hız avantajından dolayı bu özellik tanımlayıcı algoritması Hızlandırılmış KAZE (AKAZE) olarak adlandırılmıştır. Bu algoritmanın sağladığı düşük hesaplama zamanı özelliğini korumak için özellik tanımlayıcı algoritma da geliştirilerek Modified Local Difference Binary (M-LDB) algoritması önerilmiştir. KAZE'de kullanılan özellik tanımlayıcı algoritma olan LDB algoritması dönmeye ve ölçeklemeye dayanıklı değil iken M-LDB algoritması ölçek ve dönmeden bağımsızdır [49, 50].

AKAZE algoritmasının SURF, SIFT ve KAZE'ye göre daha hızlı olduğu ve ayrıca ORB ve BRISK de dâhil olmak üzere daha önceden önerilen bütün anahtar çıkarma yöntemlerine göre de daha etkin çalıştığı belirtilmiştir [50].

Doğrusal olmayan dağılım filtreleme, görüntünün parlaklığının artan ölçek seviyeleri boyunca dağılım işleminin aşamalarını tanımlar. Dağılım işlemi bir akış fonksiyonunun diverjansı tarafından kontrol edilir. Bu yaklaşım da doğrusal olmayan kısmi diferansiyel denklemler ile çözülür çünkü diferansiyel denklemlerin doğrusal olmayan doğası gereği görüntü parlaklığı doğrusal olmayan ölçek uzayı boyunca dağılmış olur. Klasik bir doğrusal olmayan diferansiyel Eşitlik (1.20)' deki gibi formüle edilebilir.

$$\frac{\partial L}{\partial t} = div(c(x, y, t).\nabla L) \tag{1.20}$$

Buradaki div diverjansı , ∇ gradyan işlemini ve L de görüntü parlaklığını göstermektedir. Bu diferansiyel denklemde iletkenlik (conductivity) fonksiyonunun tanımlanmasıyla birlikte dağılımın yerel görüntü yapısına uyarlanabilir olması mümkün olmaktadır. $c(x, y, t)$ fonksiyonu yerel görüntünün diferansiyel yapısına dayanmaktadır. Zamanı ifade eden t değeri ise ölçek parametresidir ve büyük t değerleri daha basit görüntüleri temsil eder. İletkenlik fonksiyonu olan $c(x, y, t)$ Eşitlik (1.21)'deki gibi tanımlanmaktadır.

$$c(x,y,t) = g(|\nabla L_\sigma(x,y,t)|) \tag{1.21}$$

Burada ∇L_σ ifadesi orijinal görüntü olan L'nin Gauss bulanıklaştırılmış halinin gradyanını temsil etmektedir. Perona ve Malik'in çalışmalarında iletkenlik fonksiyonun iki farklı formülasyonu sunulmuştur [51]. Yazarlar bu fonksiyonlardan daha geniş bir bölgeyi sağlayan (1.22)'de verilen g_2 fonksiyonunu kullanmışlardır [50].

$$g_2 = \frac{1}{1+\frac{|\nabla L_\sigma|^2}{\lambda^2}} \tag{1.22}$$

Buradaki kontrast parametresi olan λ dağılımın seviyesini kontrol eder ve yok edilecek veya varlığı korunacak kenarların belirlenmesini sağlar. $|\nabla L_\sigma|$' nin histogramı olan H, kontrast parametresinin belirlenmesi için oluşturulur. Histogramın *%70*' lik kısmı uygun λ kontrast parametresini seçmek için kullanılır. Doğrusal olmayan dağılım denklemini çözecek analitik bir çözüm olmadığı için kısmi diferansiyel denklemi ayrıştıracak bir çözüm tahmin etmek gerekmektedir. Bunun için yazarlar Fast Expilicit Diffusion (FED) şemasını kullanmışlardır [50].

1.4.4.1. FED (Fast Explicit Diffusion)

FED' in temel mantığında n adet dağılım adımının, τ_j değişen adım büyüklüğü ile birlikte M adet döngüde icra edilmesi vardır. Eşitlik (1.23)'de verilen τ_j kutu filtrelerinin çarpanlara ayrılmasından gelmektedir.

$$\tau_j = \frac{\tau_{max}}{2cos^2\left(\pi\frac{2j+1}{4n+2}\right)} \tag{1.23}$$

τ_{max} maksimum adım büyüklüğü 0.25 değeridir ve kararlılığı değişmez. Bir FED döngüsünün durma zamanını ifade eden θ_n değeri Eşitlik (1.24)' deki gibi elde edilir.

$$\theta_n = \sum_{j=0}^{n-1} \tau_j = \tau_{max}\,\frac{n^2+n}{3} \tag{1.24}$$

Eşitlik (1.24)'in ayrıştırmasıyla vektör-matris notasyonu Eşitlik (1.25)'deki gibi elde edilir.

$$\frac{L^{i+1}-L^{i}}{\tau} = A(L^{i})L^{i} \tag{1.25}$$

Burada $A(L^{i})$ görüntünün iletkenliğini kodlayan matristir ve τ ifadesi de $\tau < \tau_{max}$ eşitliğini sağlayan sabittir.

$$L^{i+1} = (I + \tau A(L^{i}))L^{i} \tag{1.26}$$

I kimlik matisidir. Bir önceki varsayımdan yola çıkarak $L^{i+1,0} = L^{i}$, n adet τ_j adım boyutlu FED döngüsü aşağıdaki gibi elde edilmektedir.

$$L^{i+1,j+1} = \left(I + \tau_j A\left(L^{i}\right)\right) L^{i+1,j}, \quad j = 1 \cdots n\text{-}1 \tag{1.27}$$

FED döngüsü boyunca $A\left(L^{i}\right)$'nin değeri sabittir ve döngü tamamlandıktan sonra $A\left(L^{i}\right)$ matrisinin yeni değerleri hesaplanmaktadır.

Doğrusal olmayan ölçek uzayının oluşturulması, anahtar noktalarının çıkarılması ve anahtar noktalarının özellik tanımlayıcılarının belirlenmesi adımları AKAZE yönteminin temel adımlarıdır.

1.4.4.2. Doğrusal Olmayan Ölçek Uzayının Oluşturulması

AKAZE doğrusal olmayan ölçek uzayını kullanarak Gauss ölçek uzayının sebep olduğu problemlerin üstesinden gelmektedir. Doğrusal olmayan ölçek uzayı sayesinde görüntü bağdaşık bir şekilde bulanıklaştırılır ve bu sayede nesne detayları korunurken gürültü azaltılır. Yazarlar doğrusal olmayan ölçek uzayını oluşturmak için bir önceki bölümde anlatılan FED matematiksel yapısından faydalanılır [50].

O oktav $\{0 \cdots O-1\}$ ve S alt seviyelerden (sub-level) $\{0 \cdots S-1\}$ oluşan ölçek uzayı oluşturulmaktadır. O oktav ve S alt seviye indekslerini temsil edecek şekilde ölçek uzayındaki görüntüler $L_1 \cdots L_{O \times S}$, orijinal görüntü olan L'den üretilmektedir. Oktav ve sublevel indeksleri bunlara karşılık gelen σ ölçeği ile aşağıdaki gibi eşlenir.

$$\sigma_i(o,s) = 2^{o+s/S}, \quad \mathrm{i} \in [0 \ldots \mathrm{M}] \tag{1.28}$$

Burada *M* filtrelenmiş görüntü sayısıdır. Daha sonra piksel birimindeki σ, ayrık ölçek seviyeleri zaman birimine Eşitlik (1.29)'daki gibi dönüştürülür.

$$t_i = \frac{1}{2}{\sigma_i}^2 \quad i \in [0 \ldots M] \tag{1.29}$$

Ayrıca girdi görüntüsü σ_0 standart sapmasıyla birlikte gürültü ve kalıntıları azaltmak için Gauss bulanıklaştırma işlemine tabi tutulabilir. Bulanıklaştırılmış girdi görüntüsünden kontrast faktörü olan λ değeri gradyan histogramının *%70'* i olacak şekilde elde edilir.

Ölçek uzayının oluşturulması verilen girdi görüntüsü, kontrast faktörü ile birlikte *M-1* kere çağırılarak FED döngüsünün icrası ile gerçekleştirilir. Her FED döngüsü için minimum adım sayısı olan *n* hesaplanır. Girdi görüntüsünün iki boyutlu olması durumunda, maksimum adım büyüklüğü τ_{max} *=0.25* olarak ele alınır. Her FED döngüsünün zamanı $T = t_{i+1}$-t_i olarak hesaplanmaktadır.

Her oktavın ilk görüntüsü $(L_1, L_{1+S}, L_{1+2S} \cdots)$ bir önceki oktavın son görüntünün faktör *2* ve bulanıklaştırma maskesi olan $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$ ile alt örneklemesi (subsampling) sonucu elde edilmektedir. Görüntünün alt örneklemesi yapıldıktan sonra da konstast parametresinin güncellenmesi gerekmektedir. Kontrast parametresi olan λ'de ince kenarları tespit edebilmek için önceki oktava ait kontrast parametresinin $t_{cp} = 0.75$ ile çarpımından elde edilmektedir. Ölçek uzayının oluşturulmasına dair algoritma Şekil 1.13'de verilmiştir. Şekil 1.14' de ise FED döngüsü için iterasyon adımları verilmiştir.

Girdi: Orijinal görüntü L^0, kontrast parametresi λ , τ_{max} ve evrim zamanı dizisi t_i

Çıktı: Filtrelenmiş görüntüler dizisi L^i i=0..M

For i=0→ M-1

1.Yayılganlık matrisinin $A(L^i)$'in hesaplanması

2. FED döngüsünün süresi $T = t_{i+1} - t_i$

3.FED adım sayısı n' in hesaplanması

4.Adım büyüklüğü τ_j 'nin hesaplanması

5. $L^{i+1,0} = L^i$

$L^{i+1} = \boldsymbol{FED}(L^{i+1,0}, A(L^i), \tau_j)$

if $o_{i+1} > o_i$

$(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$ maskesiyle birlikte L^{i+1} görüntüsünün aşağı örneklendirilmesi

$\lambda = \lambda \, x \, 0.75$

end if

end for

Şekil 1.13. Doğrusal olmayan ölçek uzayının oluşturulması algoritması

function $\boldsymbol{FED}$ $(L^{i+1,0}, A(L^i), \tau_j)$

for j=0→n-1

$L^{i+1,j+1} = (I + \tau_j A(L^i))L^{i+1,j}$

end for

Return $L^{i+1,n}$

end function

Şekil 1.14. FED döngüsü algoritması

1.4.4.3. Anahtar Noktalarının Çıkarılması

Oluşturulan doğrusal olmayan ölçek uzayındaki her görüntü için bir anahtar noktası çıkarma işlemi gerçekleştirilir. Her görüntünün Hessian matrisi hesaplanır ve normalize

edilmiş ölçek faktörüyle çarpılır. Bu ölçek faktörü her görüntü için farklıdır. L^i görüntüsü için ölçek faktörü sf_i , Eşitlik(1.30)' daki gibi hesaplanmaktadır.

$$sf_i = \sigma_i / 2^{o^i} \tag{1.30}$$

$$L^i_{Hessian} = {sf_i}^2 (L^i_{xx} L^i_{yy} - L^i_{xy} L^i_{xy}) \tag{1.31}$$

Daha sonra ölçek uzayındaki extremum noktaları bulmak içim ölçeklendirilmiş Hessian matrisinin determinantını hesaplanmaktadır. Matris determinantının önceden tanımlanmış bir eşik değerinden büyük olup olmadığı kontrol edilir. Eğer büyük ise bu nokta *3x3* komşuluğunda maksimum nokta olduğu ortaya konmaktadır. Bu şekilde maksimum olmayan noktaların hızlı bir şekilde eliminasyonu gerçekleşmektedir. Ardından her potantiyel maksimum noktası için sırasıyla hizasındaki bir üst seviyesine ve hizasındaki bir alt seviyesine, bakılarak noktanın maksimum olup olmadığı tekrar kontrol edilir. Eğer daha büyük olduğu tespit edilirse bu noktanın anahtar noktası olduğu anlaşılmaktadır.

1.4.4.4. Özellik Tanımlayıcının Çıkarılması

AKAZE bir önceki adımda belirlenen anahtar noktalarına ait özellik tanımlayıcısını elde etmek için LDB algoritmasını ölçekleme ve dönmeye karşı bağımsız bir sonuç üretecek şekilde modifiye ederek kullanmıştıt [50]. LDB özellik çıkarma algoritması [52], BRIEF [45] algoritmasına benzer bir yaklaşım kullanmaktadır. Ancak LDB algoritmasındaki gibi her piksel için ikili karşılaştırmalar yapılması yerine bölgenin ortalamasını kullanarak karşılaştırma yapılmaktadır. İlgili bölgenin yatay ve dikey türevlerinin ortalamaları ayrıca karşılaştırma için kullanılmaktadır. Bu şekilde üç bit gösterimli ikili karşılaştırma sonucu elde edilmektedir.

LDB ilgili alanı *2x2, 3x3* gibi farklı boyutlu alanlara bölünmektedir. İntegral görüntüler kullanılmasıyla bu alt bölgelerin ortalamalarının hesaplanması hızlı bir şekilde gerçekleştirilmektedir. Ancak integral görüntülerin kullanılması özellik çıkarma metodunu dönme ataklarına karşı savunmasız hale getirmektedir. Bu yüzden AKAZE, LDB algoritmasını dönmeye karşı duyarlı hale getirmek için ana dönme bilgisini

kullanmaktadır. M-LDB, LDB çerçevesini ana dönme bilgisine göre döndürmektedir. Aşağıda algoritmanın nasıl çalıştığına dair adımlar verilmiştir:

1.Adım: Anahtar noktasının ana dönme bilgisi SURF algoritmasına benzer şekilde belirlenir [41].

2.Adım: Alt örnekleme adım büyüklüğü, algoritmada kullanılan örüntü(pattern) büyüklüğüne göre belirlenir. Örüntü büyüklüğü *12*'dir ve adım büyüklüğü $\{5, \lceil 5(2/3) \rceil = 4, \lceil 5(1/2) \rceil = 3\}$' dir. Anahtar noktası oktav değeri ile ölçeklendirilir ve (k_x, k_y) olarak gösterildiği varsayılırsa (k_x, k_y) etrafında *12x12*' lik pencere *5x5*, *4x4* ve *3x3* boyutlu 2x2, 3x3 ve 4x4' lük alt bölgelere bölünür. Her alt bölge ana dönme bilgisine göre döndürülür ve ortalama piksel değerleri, yatay ve dikey yöndeki türevleri hesaplanır. Bu bağlamda *12x12'* lik pencere adım büyüklüğü 5, 4 ve 3 olan *2x2*, *3x3* ve *4x4'* lük alt bölgelerden sırasıyla *12(2x2x3)*, *27(3x3x3)* ve *48(4x4x3)* adet ortalama değer elde edilir. Elde edilen bu değer de geçici *1x87* boyutundaki *T* vektörüne eklenir.

3.Adım: Bu adımda *T* vektöründeki elemanlardan aynı adım büyüklüğünden elde edilenlerin kendi aralarında 3 bitlik bir karşılaştırması yapılmaktadır. Çerçeve adedi Ç olarak gösterildiğinde *(Ç-1)*(Ç)/2* adet üçer bit ikili karşılaştırması yapılmaktadır. Yani *2x2*'den elde edilen vektörün Ç=*4* olduğundan adet üç bitlik değer için ((4-1)*4)/2=6 adet karşılaştırma yapılarak *18* bit uzunluğunda sonuç elde edilir ve bu da özellik tanımlayıcı vektörün 1-18 değerleri arasını oluşturmaktadır. Benzer şekilde *3x3* ve *4x4'* lük kısımlardan sırasıyla 108 bit ve *360* bitlik ikili karşılaştırma sonuçları üretilir. *18+108+360=486* olmak üzere *1x486* boyutunda özellik tanımlayıcı vektör elde edilmiş olur.

1.5. Doku Çıkarma Yöntemleri

Çalışma kapsamında önerilen yöntemlerde test görüntüsünden anahtar noktası elde etmek yerine, test görüntüsünün doku görüntüsü üzerinden anahtar nokta çıkarımı ve eşleştirmesinin daha iyi sonuçlar ürettiği gözlemlenmiştir. Tez kapsamında uygulanan yöntemlerde yer alan LPQ ve Gabor filtresi doku çıkarma yöntemleri bu bölümde incelenecektir.

1.5.1. Yerel Faz Nicemleme (Local Phase Quantization, LPQ)

LPQ, Ojansivu ve Heikkila tarafından 2008 yılında bir doku tanımlayıcısı olarak önerilmiştir [46]. LPQ iki boyutlu Kıza Zamanlı Fourier Dönüşümü(Short Time Fourier Transform, STFT)'nü kullanarak görüntünün yerel faz bilgisini elde eder. Bu bilgi her piksel için bir pencerede yerel olarak hesaplanmaktadır.

Uzaysal domainde resmin bulanıklaştırma işlemi görüntünün Noktasal Yayılım Fonksiyonu (Point Spread Function, PSF) sonucu ve görüntünün yoğunluk bilgisinin konvolüsyonu işlemi ile gerçekleştirilmektedir. Orijinal görüntü *f(x)*, bulanıklaştırılmış görüntü *g(x)*, ve *h(x)*'de bulanıklaştırılmış görüntünün PSF'si olarak ele alındığında, bu konvolüsyon işlemi Eşitlik (1.32)'deki gibi gerçekleştirilir.

$$g(x) = f(x) * h(x) \tag{1.32}$$

(1.32)'de verilmiş olan x ,$[x, y]^T$ koordinatların bir vektörüdür. Bu eşitlik Fourier domeninde aşağıdaki formülle ifade edilir.

$$G(u) = F(u) * H(u) \tag{1.33}$$

(1.33)'de u , $[u, v]^T$ koordinatlarının frekans vektörüdür. Frekans domenindeki bu konvolüsyonun büyüklüğü ve fazı (1.34)'deki ifade ile elde edilir.

$$|G(u)| = |F(u)|.|H(u)| \text{ ve } \angle G(u) = \angle F(u) + \angle H(u) \tag{1.34}$$

Bulanıklaştırılmış görüntünün PSF' si olan *h(x)* merkezi simetrik özelliğe sahip olduğundan *h(x)=h(-x)* denilebilir. Ayrıca bu değerin Fourier transformu her zaman gerçek değerlidir. Dolayısıyla *H(u)* aşağıdaki durumlar koşulunda sadece iki değer içerir $\angle H(u) \in (0,\pi)$.

$$\angle H(u) = \begin{cases} 0 \rightarrow H(u) \geq 0 \\ \pi \rightarrow H(u) < 0 \end{cases} \tag{1.35}$$

LPQ'da *f(x)* görüntüsü için her *x* pikselin yerel komşuluğu *Nx'* de faz analizi yapılır. Alçak frekans çözümlemelerinin yüksek uzaysal çözümlemelere karşılık geldiğinden düşük frekans faz açısı merkezi simetrik bulanıklaştırmayı sağlar. Bu yerel spektrumlar yani resimdeki her bir pikselin yerel komşuluk faz değerleri STFT kullanarak Eşitlik (1.36)'daki gibi hesaplanır.

$$F(u,x) = \sum_{y \in N_x} f\,(x-y) e^{-j 2\pi u^T y} \tag{1.36}$$

Görüntünün satır ve sütunlarına bir boyutlu konvolüsyon uygulanarak tüm pikseller $x \in \{ x_1, x_2, \ldots x_N \}$ için fourier faz katsayıları elde edilir. Elde edilen bu yerel Fourier katsayıları $[0, \pi/2, \pi, 3\pi/2]$ açıları için hesaplanmaktadır. Bu açılarda iki boyutlu $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u_3 = [a, a]^T$, ve $u_4 = [a, -a]^T$ frekanslarına denk gelmektedir. Buradaki *a*, pencere boyutu olan *m* den, *a =1/m* eşitliği kullanılarak elde edilir. Her piksel için yerel Fourier katsayıları elde edilen iki boyutlu frekanslara karşılık gelen katsayılarla Eşitlik (1.37)' deki gibi kuantalanır.

$$F_x^c = [\{F(u_1, x), F(u_2, x), F(u_3 x), F(u_4, x)\}] \tag{1.37}$$

$$F_x = [Re\ \{F(x), Im\{F(x)\}]^T \tag{1.38}$$

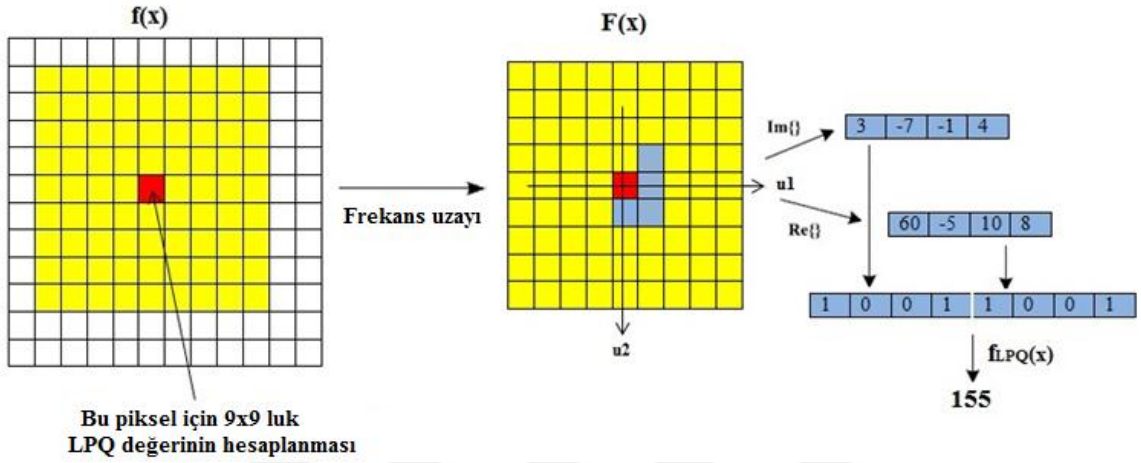
(1.38)'de *Re{·}* ve *Im{·}*, elde edilen karmaşık sayının sırasıyla reel ve sanal bölümlerini vermektedir. Daha sonra , G_x hesaplanır ve sonuç vektörü Eşitlik (1.39)' daki gibi eşiklenir.

$$q_j = \begin{cases} 1 \quad \rightarrow \ g_j \geq 0 \\ \\ 0, \text{ diğer durumda} \end{cases} \tag{1.39}$$

(1.39)'daki g_j , $G(x) = [Re\ \{F(x), Im\{F(x)\}]$ vektörünün *j*. elemanıdır. Son olarak x noktasının yerel yapısını temsil eden $f_{LPQ}(x)$ Eşitlik (1.40)'daki gibi elde edilir. Bu değer 8 bitlik 0-255 arası ikili bir koddan 8-bitlik ikilik kodun onluk (decimal) değeri bulunmasıyla elde edilmektedir.

$$f_{LPQ}(x) = \sum_{j=1}^{8} q_j \, 2^{j-1} \tag{1.40}$$

Şekil 1.15' de pencere boyutu olan m değerinin 9 olarak seçilmesi durumunda $f_{LPQ}(x)$ değerinin hesaplanması işleminin özeti verilmiştir.



Şekil 1.15. LPQ yönteminin özeti

1.5.2. Gabor Filtresi

Gabor filtresi görüntü işlemede görüntü alma ve doku analizinde yaygın olarak kullanılan bir doğrusal filtredir [48]. Gauss filtresi Gauss fonksiyonun karmaşık sinüs sinyali ile yumuşatılması ile Eşitlik (1.41)' deki gibi hesaplanmaktadır.

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right).e^{\left[-\frac{1}{2}\left(\frac{x'^2}{\sigma_x{}^2}+\frac{y'^2}{\sigma_y{}^2}\right)\right]}.e^{(2\pi jWx)} \tag{1.41}$$

Buradaki x' ve y' değerleri (1.42)'deki gibi hesaplanmaktadır.

$$\begin{aligned} x' &= \quad xsin\theta + ycos\theta \\ y' &= -xcos\theta + ysin\theta \end{aligned} \tag{1.42}$$

σ_x ve σ_y değerleri Gauss fonksiyonunun x ve y ekseni boyunda standart türevleridir. W değeri sinüs sinyalinin merkezi sıklığını ve θ'de dönme açısını vermektedir. Düz (smooth) görüntünün Gabor temsili Eşitlik (1.43)' de de verildiği gibi görüntünün Gabor filtresiyle konvolüsyonuna eşittir.

$$G_{f,\theta}(x,y) = I(x,y) * g_{f,\theta}(x,y) \tag{1.43}$$

Buradaki * işlemi konvolüsyon operatörünü, f uzaysal frekansı, $I(x,y)$ gri seviyeye çevrilen görüntünün *(x,y)* koordinatlarındaki yoğunluk bilgisini temsil etmektedir. Eşitlik (1.43)' de verilen formül ile görüntüye değişen dalga boyu ve yönelim bilgisi uygulanmaktadır.

1.6. Özellik Noktası Eşleştirme

Bu kısımda tez kapsamında SIFT, SURF, ORB ve AKAZE algoritmaları kullanılarak elde edilen özellik vektörlerinin eşleştirilmesi için, özellik vektörlerinin mutlak uzaklıklarının hesaplanmasında kullanılan metrikler verilmiştir.

1.6.1. Öklid Uzaklığı

Öklid uzaklığı, $p = (p_1, p_2, \dots, p_n)$ ve $q = (q_1, q_2, \dots, q_n)$ şeklinde verilen iki rassal p ve q vektörleri arasındaki doğrusal uzaklık olarak tanımlanabilir. Öklid uzaklığı olan $d_{Öklid}$ Eşitlik (1.44)'deki gibi hesaplanmaktadır.

$$d_{Öklid} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{1.44}$$

1.6.2. Hamming Uzaklığı

Hamming uzaklığı aynı uzunluktaki iki dizenin farklı elemanlarının sayısıdır. $p = (p_1, p_2, \dots, p_n)$ ve $q = (q_1, q_2, \dots, q_n)$ vektörlerinin Hamming uzaklığı olan $d_{Hamming}$ Eşitlik (1.45)' deki gibi hesaplanmaktadır.

$$d_{Hamming} = \sum_{i=1}^{n} XOR(p_i - q_i) \tag{1.45}$$

1.7. RANSAC (Random Sample Consensus)

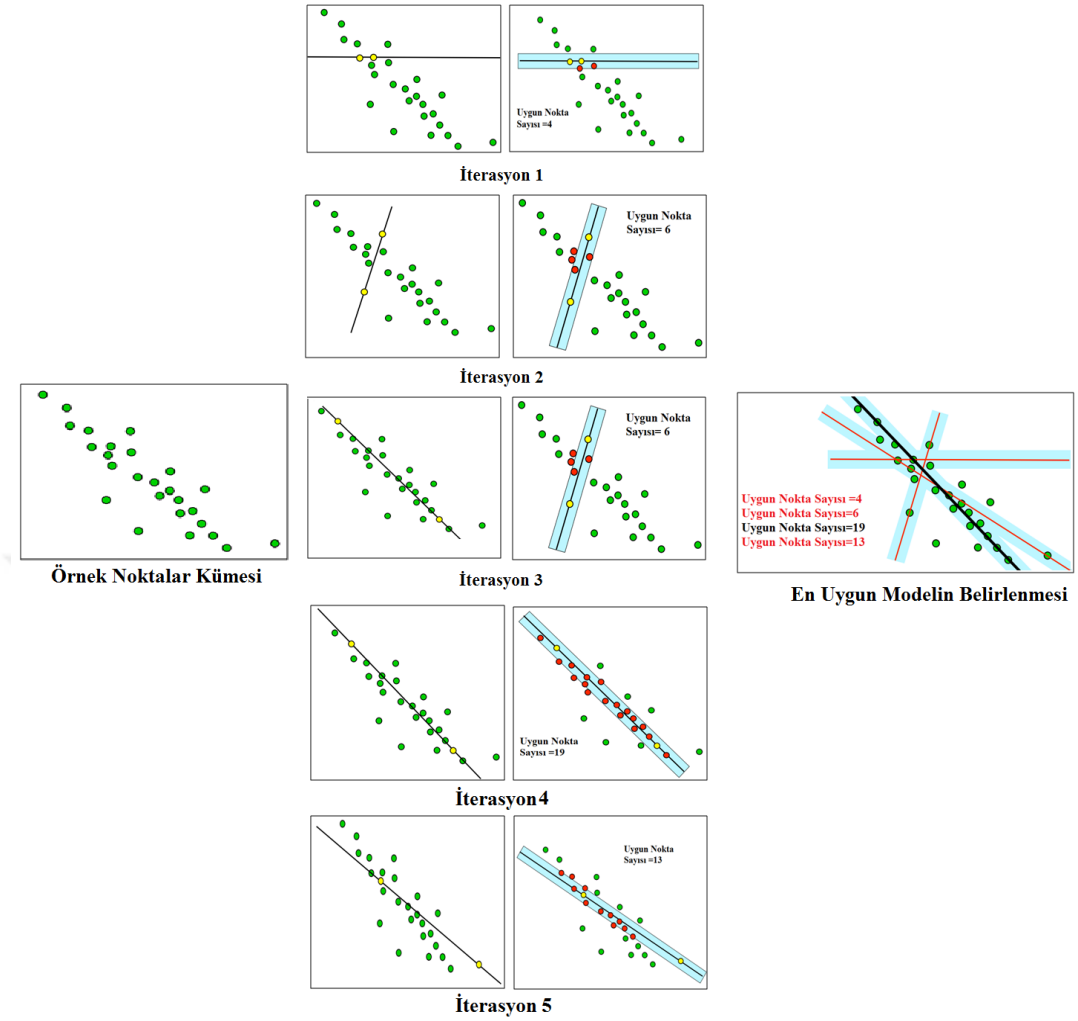
Tez çalışması kapsamında yapılan çalışmalarda anahtar noktalarının eşleştirilmesi işleminden sonra varsa hatalı eşleştirmelerin yok edilmesi için RANSAC algoritması kullanılmış olup, bu algoritma bu kısımda incelenecektir.

RANSAC (Random Sample Consensus) yüksek oranda yanlış eşleşmelere sahip veri setindeki hataları minimize etmek için Fischler tarafından önerilen tekrarlamalı bir yöntemdir [48]. Model parametrelerini tahmin etmek için gerekli olan minimum sayıda gözlem noktası (veri) içeren kümeyi kullanarak, tutarlı veri noktalarıyla bu kümeyi genişletip aday çözümler üretir. Bu yöntemde rastgele eşleşen belli sayıda anahtar noktaları seçilerek (1.46)'da verilen H transformasyon matrisinin parametreleri hesaplanmaktadır.

$$H\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} \tag{1.46}$$

Transformasyon matrisi parametreleri ile eşleşen anahtar noktaları arasındaki Öklid uzaklık değeri hesaplanmaktadır. Bu uzaklık değerinin önceden tanımlanan γ eşik değerinden küçük olması durumunda eşleşen noktaları doğru eşleşme(inlier) olarak ifade edilir. Uzaklık değerinin γ eşik değerinden büyük olması durumunda (aykırı durum) ise bu eşleşen noktalar hatalı eşleşme (outlier) olarak kabul edilir ve M eşleşme matrisinden çıkartılır.

Şekil 1.16' da örnek bir noktalar kümesine ait beş iterasyonda en uygun modelin belirlenmesi işlemi özetlenmiştir. Burada rastgele seçilen nokta sayısı *2* olarak alınmıştır. Uzaklığın kontrolünde eşik değeri γ*=0.5* olarak kabul edilmiştir. Görsel gösterim için rastgele noktalardan geçen bir doğrunun (model) etrafında eni 1 birim olan mavi renkli bir dikdörtgen çizilmiştir. Bu dikdörtgenin sınırlarına giren noktaların uzaklık şartını sağladığı kabul edilmektedir. İlk iterasyonda rastgele seçilen iki noktaya göre uygun nokta sayısı 4 iken ikinci iterasyonda nokta sayısı 6'dır. Benzer şekilde diğer iterasyonlar için de uygun nokta sayıları hesaplanır. En fazla uygun nokta sayısını sağlayan doğru(model) uygun model olarak seçilir. Bu doğruya uzaklığı *0.5* birim olmayan noktalar outlier olarak belirlenmiştir.



Şekil 1.16. RANSAC ile örnek tekrarlamalı olarak uygun modelin belirlenmesi

2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME

2.1. Giriş

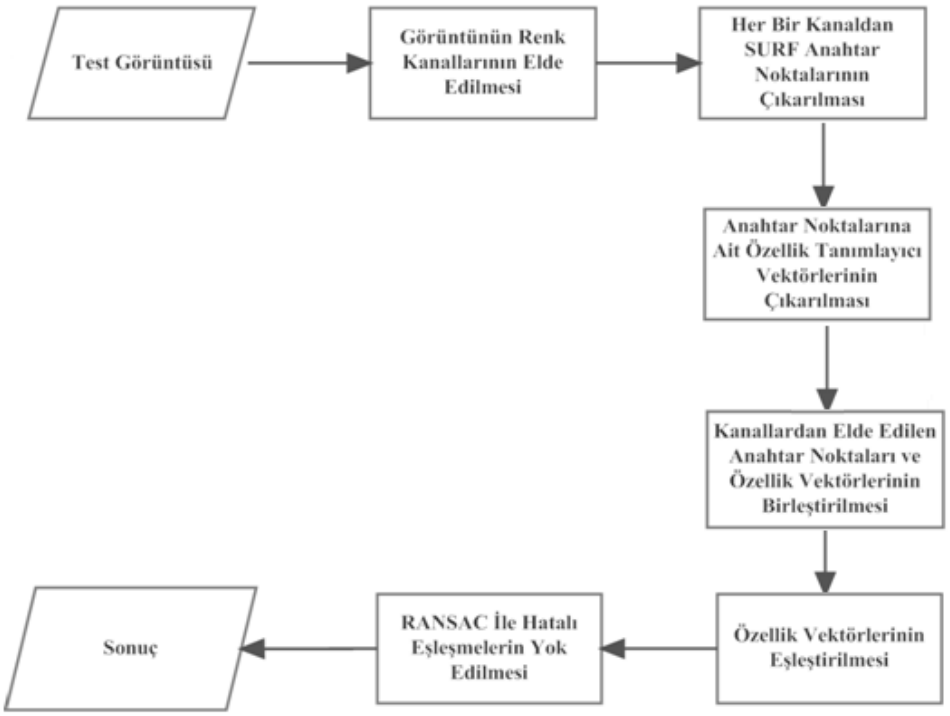
Bu tez çalışmasında anahtar noktası tabanlı kopyala yapıştır sahteciliği tespitine ilişkin literatürde var olan anahtar noktası tabanlı yöntemler incelenerek [34, 35, 38]'deki çalışmalar iyileştirilmiş olup ayrıca yeni ve etkin bir anahtar noktası tabanlı yöntem önerilmiştir.

İlk çalışma olarak SURF tabanlı [34]'de önerilen yöntemin irdelenmesi ve iyileştirilmesi gerçekleştirilmiştir. Ancak bu yöntemin yapılan iyileştirme sonrasında bile düz (smooth) bölgelerde başarısız olduğu gözlemlenmiştir. Anahtar noktası tabanlı kopyala yapıştır sahteciliği tespiti yapan SIFT ve ORB tabanlı [35,38]'deki çalışmalarda da aynı problemle karşılaşılmıştır. Bunun sebebi ise düz bölgelerde çok az sayıda ve hatta hiç anahtar noktası bulunamaması olarak ortaya konulmuştur.

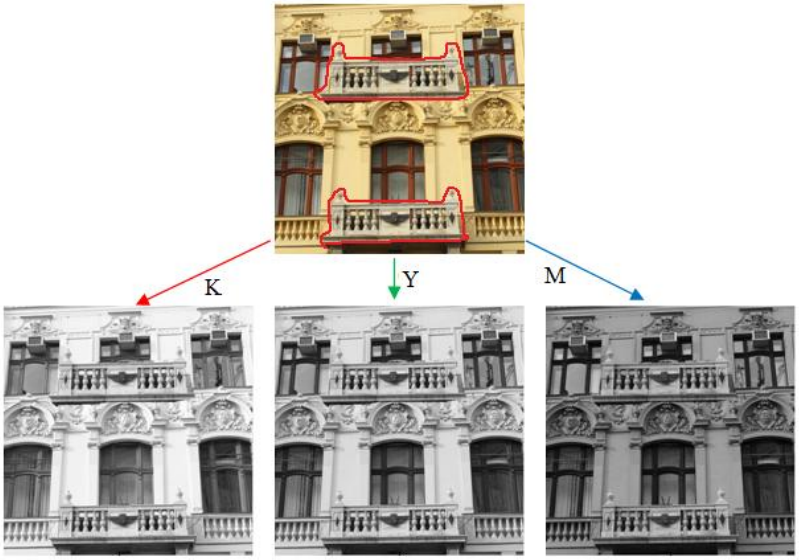
Anahtar noktası tabanlı kopyala yapıştır sahteciliğinde; anahtar noktası çıkarması olarak SIFT, SURF ve ORB tabanlı yöntemlerde Gauss ölçek uzayı oluşturulmaktadır[31-38]. Gauss ölçek uzayının önemli bir dezavantajı olarak nesne sınırlanırının korunamaması, gürültü ve detayların bütün ölçekler boyunca aynı derecede bulanıklaştırılması gösterilebilir. Bu durumda düz bölgelerde anahtar noktası tespitinin yapılamamasına neden olmaktadır. Daha sonraki çalışmalarda bu problemin çözümlenmesi amaçlanmıştır. Bu amaçla düz bölgelerde daha etkin çalışabilmesi için görüntüden öncelikle doku bilgisi elde edilerek daha sonra kopyala yapıştır sahteciliği tespiti yapılması önerilmiştir. Yapılan ilk çalışmada görüntüden önce LPQ ile doku bilgisinin elde edilmesi ve ardından SIFT anahtar noktası elde edilerek kopyala yapıştır sahteciliği tespiti yapılması gerçekleştirilmiştir. Diğer bir çalışmada ise Gabor filtresi ve ORB tabanlı yeni bir sahtecilik tespiti yöntemi önerilmiştir. Tez kapsamında gerçekleştirilen son çalışmada ise görüntülerden doku çıkarmadan da, düz bölgeler aracılığıyla uygulanan kopyala yapıştır sahteciliğinin mümkün olmasını sağlayan AKAZE tabanlı yeni bir anahtar noktası tabanlı kopyala yapıştır sahteciliği tespiti yöntemi ile belirlenmesi sağlanmıştır. Önerilen bu yöntem hem düz bölgelerde hem de karmaşık bölgelerle yapılan sahteciliklerin tespitini gerçekleştirmektedir.

2.2. Renkli SURF ile Anahtar Noktası Tabanlı Kopyala Yapıştır Sahteciliği Tespiti

Literatürdeki SURF tabanlı kopyala yapıştır sahteciliği tespiti yöntemleri görüntünün gri seviyeye dönüştürülmesi (R*0.3+G*0.59+B*0.11) ön işleminden sonra çalışmaktadır. Gri seviyeye dönüştürülen görüntüden SURF algoritması ile anahtar noktası ve bu anahtar noktaları elde edilmekte ve bu anahtar noktalarının özellik tanımlayıcıları üretilmektedir [34, 37]. Yapılan çalışmada *NxMx3* büyüklüğündeki renkli sahte görüntü ilk olarak Şekil 2.2' deki gibi kırmızı yeşil ve mavi renk kanallarına ayrılmakta ve bu kanalların her birine SURF algoritması uygulanmaktadır. Böylece bu üç kanaldan *k, y, m* adet anahtar noktası elde edilmiş olur. Daha sonra elde edilen anahtar noktalarının ve özellik tanımlayıcı vektörlerinin birleştirilmesi gerçekleştirilmektedir. Böylece anahtar nokta sayısı kadar *64* boyutlu özellik tanımlayıcıları elde edilir ve *D=(k+y+m)x64* boyutlu bir matriste tutulur. Daha sonra bu özellik tanımlayıcı vektörlerinin eşleştirilmesi gerçekleştirilmiştir. Eşleşme işleminden sonra hatalı eşleşmelerin yok edilmesi için RANSAC algoritması kullanılmıştır. Şekil 2. 1' de yapılan çalışmaya ait blok diyagram verilmiştir.

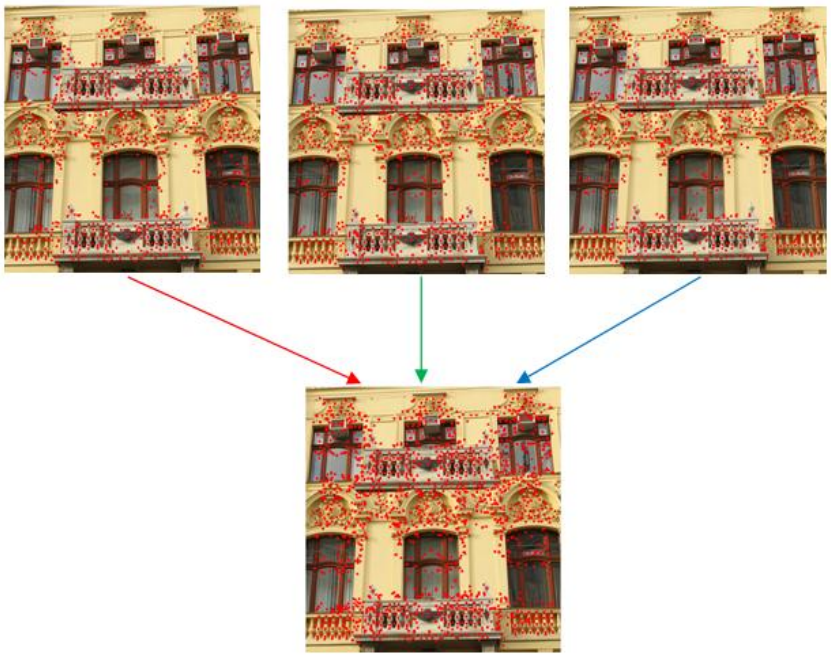


Şekil 2. 1. Yapılan renkli SURF tabanlı çalışmanın blok diyagramı



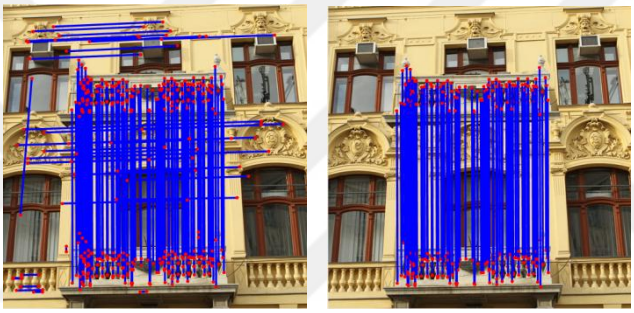
Şekil 2. 2. Sahte renkli görüntünün RGB renk kanallarına ayrılması

Şekil 2. 2'deki örnek sahte görüntüye ait ayrılan kırmızı yeşil ve mavi renk kanallarından sırasıyla SURF algoritmasıyla *1195*, *1284*, *1238* adet anahtar noktası çıkarılmıştır. Şekil 2. 3' de her bir kanaldan elde edilen anahtar noktaları görülmektedir. Bu anahtar noktalarına ait *64* boyutlu özellik tanımlayıcı da çıkarılmıştır. Bu özellik tanımlayıcılarının birleştirilmesiyle *3717x64*'lük bir özellik vektörü elde edilmiştir.



Şekil 2. 3. Her bir renk kanalından elde edilen anahtar noktalarının ve özellik vektörlerinin birleştirilmesi

Eşleşme işleminde her bir özellik vektörünün diğer özellik vektörleri ile arasındaki Öklid uzaklığı hesaplanır ve sıralı $D=\{d_1, d_2, \dots, d_{n-1}\}$ uzaklık vektörleri oluşturulur. Her bir f_i vektörü için $d_1/d_2 > T, T \in (0,1)$ ise bu iki özellik tanımlayıcısının ait olduğu anahtar noktalar eşleştirilir. Değerler arasındaki bu oran T eşik değerinden büyük olduğunda ise bir eşleşme yapılmamaktadır. Yapılan çalışmada T eşik değeri *0,6* olarak belirlenmiştir. Şekil 2. 3'de elde edilen *3717x64* boyutlu özellik vektörünün eşleşme sonucu Şekil 2. 4 (a)' da verilmiştir. Burada görülen yanlış eşleşmeler RANSAC algoritması kullanılarak yok edilmiştir. RANSAC kullanımında uzaklık eşik değeri parametresi *0.001* alınmış ve eşleşmelerin uygunluğu tespit edilmiştir. RANSAC algoritmasının kullanımı ile birlikte hatalı eşleşmeler elenmiş ve Şekil 2. 4 (b)'deki sonuç görüntü elde edilmiştir.

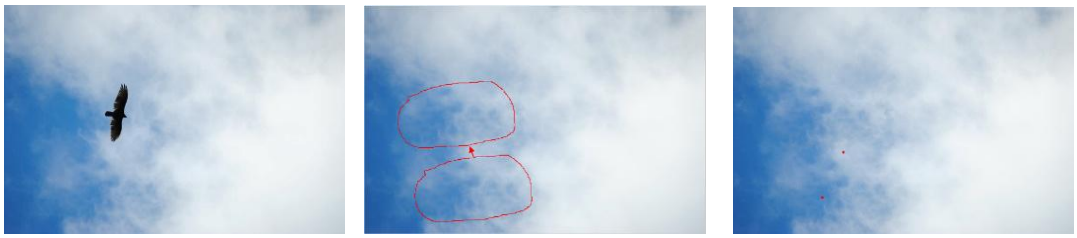


(a) (b)

Şekil 2. 4. (a) RANSAC öncesi eşleştirme sonucu (b) RANSAC ile hatalı eşleşmelerin yok edilmesi

2.3. LPQ ve SIFT Tabanlı Kopyala Yapıştır Sahteciliği Tespiti Yöntemi

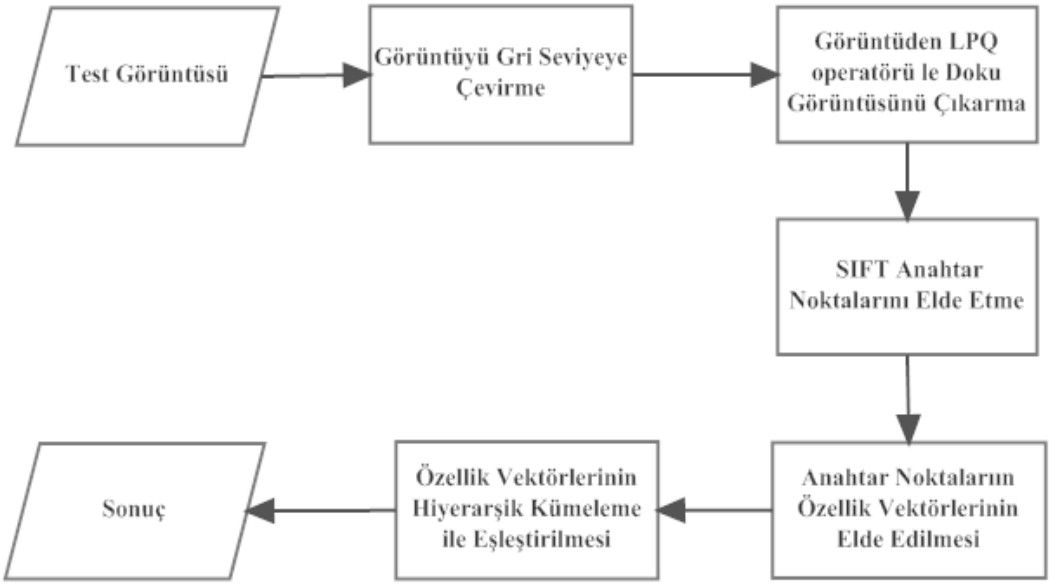
Amerini tarafından önerilen SIFT tabanlı çalışmada [35] JPEG sıkıştırma, gürültü ekleme, dönme ve ölçekleme gibi ataklara dayanıklı etkin bir anahtar noktası tabanlı kopyala yapıştır sahteciliği tespiti yöntemi önerilmiştir. Ancak SIFT algoritmasında Gauss ölçek uzayını oluştururken gerçekleştirilen Gauss bulanıklaştırma işlemi nesne kenarlarını koruyamadığı için özellikle düz bölgelerle gerçekleştirilen sahtecilik sonuçlarında anahtar noktası bulamamaktadır. Dolayısıyla anahtar noktası bulunamayacak türden bir bölge ile bir sahtecilik işlemi gerçekleştirildiğinde bu sahteciliğin tespiti [35]'de önerilen yöntem ile yapılamayacaktır. Şekil 2. 5' de kopyala yapıştır sahteciliği uygulanmış sahte görüntü olan (b)'ye [35]' de önerilen yöntem uygulanmış ve sadece *2* tane anahtar noktası tespit edilmiş olup bunlarında eşleşmesi gerçekleştirilememiştir.



(a) (b) (c)

Şekil 2. 5. (a) Orijinal görüntü (b) Sahte görüntü (c) [36]'da önerilen yöntem sonucu

Belirlenen bu problemin çözümüne ilişkin görüntünün doku bilgisinin LPQ [46] operatörüyle elde edilmesinin ardından SIFT anahtar noktalarının çıkartılarak bu anahtar noktalarına ait özellik vektörlerinin hiyerarşik kümeleme yöntemiyle eşleştirilmesi gerçekleştirilmektedir. Önerilen yöntemin blok diyagramı Şekil 2. 6' daki gibidir.



Şekil 2. 6. LPQ ve SIFT tabanlı önerilen yöntemin blok diyagramı

2.3.1. LPQ İle Doku Çıkarma

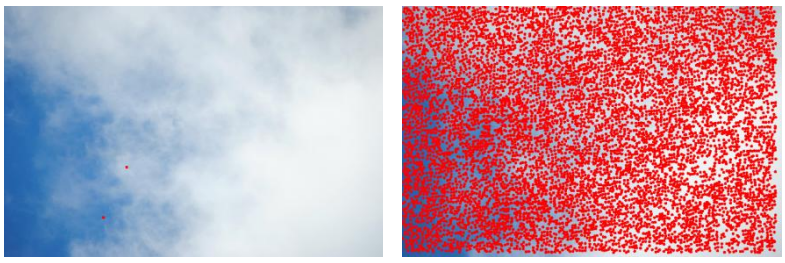
Bu çalışmada görüntünün doku bilgisi Bölüm 1.6.1'de anlatılan LPQ yöntemiyle elde edilmiş ve pencere boyutu $m=9$ olarak seçilmiştir.

(a) (b)

Şekil 2. 7. (a) Sahte görüntü (b) Sahte görüntüye ait LPQ ile elde edilen doku görüntüsü

2.3.2. SIFT ile Anahtar Noktalarının Tespiti ve Özellik Tanımlayıcı Vektörlerinin Çıkarılması

Bu aşamada sahte test görüntüsüne ait doku görüntüsü elde edildikten sonra Bölüm 1.4.1' de anlatılan SIFT algoritması ile bu görüntüden anahtar noktası çıkarma işlemi gerçekleştirilmektedir. Böylece özellikle düz bölgelerde çok az tespit edilen anahtar nokta sayısı artırılmış olur. Şekil 2. 7 (a)' daki örnek sahte görüntüden SIFT ile elde edilen 2 tane anahtar nokta Şekil 2. 8 (a)'da kırmızı ile boyanarak verilmiştir. Şekil 2. 7 (b)'deki LPQ doku görüntüsünden SIFT ile elde edilen anahtar sayısı ise *11614* tanedir ve Şekil 2. 8 (b) verilmiştir. Böylece kopyalanıp yapıştırılan bölgeye ait daha fazla anahtar noktası elde edildiğinden bu bölgelerin tespitinin yapılması daha net gerçekleştirilebilecektir.

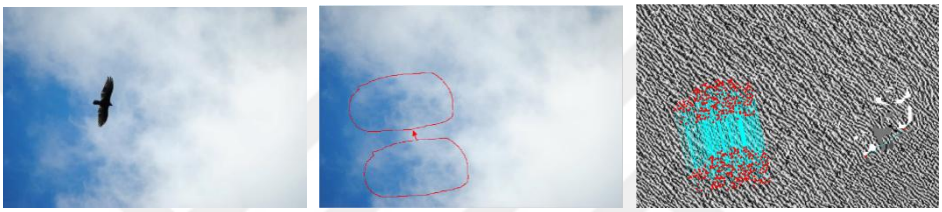


(a) (b)

Şekil 2. 8. (a) Sahte görüntüden elde edilen SIFT anahtar noktaları Anahtar nokta sayısı:2) (b) Doku görüntüsünden elde edilen SIFT anahtar noktaları (Anahtar nokta sayısı:11614)

Bir sonraki aşamada elde edilen anahtar noktalarına ait *128* boyutlu özellik tanımlayıcı vektörlerinin, $\{f_1 \dots f_n\}$, elde edilmesi gerçekleştirilmektedir. Bu özellik vektörlerinde anahtar noktasına ait koordinat bilgisi de yer almaktadır. Elde edilen bu vektörlerin eşleştirme işlemi [35]' de önerilen hiyerarşik kümeleme yaklaşımı kullanılarak

gerçekleştirilmektedir. Bu yaklaşıma göre mevcut f_i özellik vektörünün diğer özellik vektörleri ile arasındaki Öklid uzaklığı hesaplanmakta ve $D=\{d_1, d_2, \dots, d_{n-1}\}$ uzaklıklar vektörü oluşturulmaktadır. f_i vektörüne en yakın özellik vektörü ile ikinci en yakın özellik vektörü arasında bir T eşik değeri ile karşılaştırma yapılır. $d_1/d_2 > T, T \in (0,1)$ şartını sağlayan özellik vektörleri eşleştirilir. Yapılan çalışmada T eşik değeri *0,6* olarak belirlenmiştir. Eşleşmesi yapılacak olan özellik vektörlerinden koordinat bilgisi çıkartılarak eşleşen koordinatların görsel olarak gösterilmesi gerçekleştirilir. Şekil 2. 9' da önerilen bu yönteme ait örnek görsel sonuç verilmiştir. Önerilen yöntem ile elde edilen *11600* adet anahtar noktasından *206* tane eşleşme bulunmuştur.

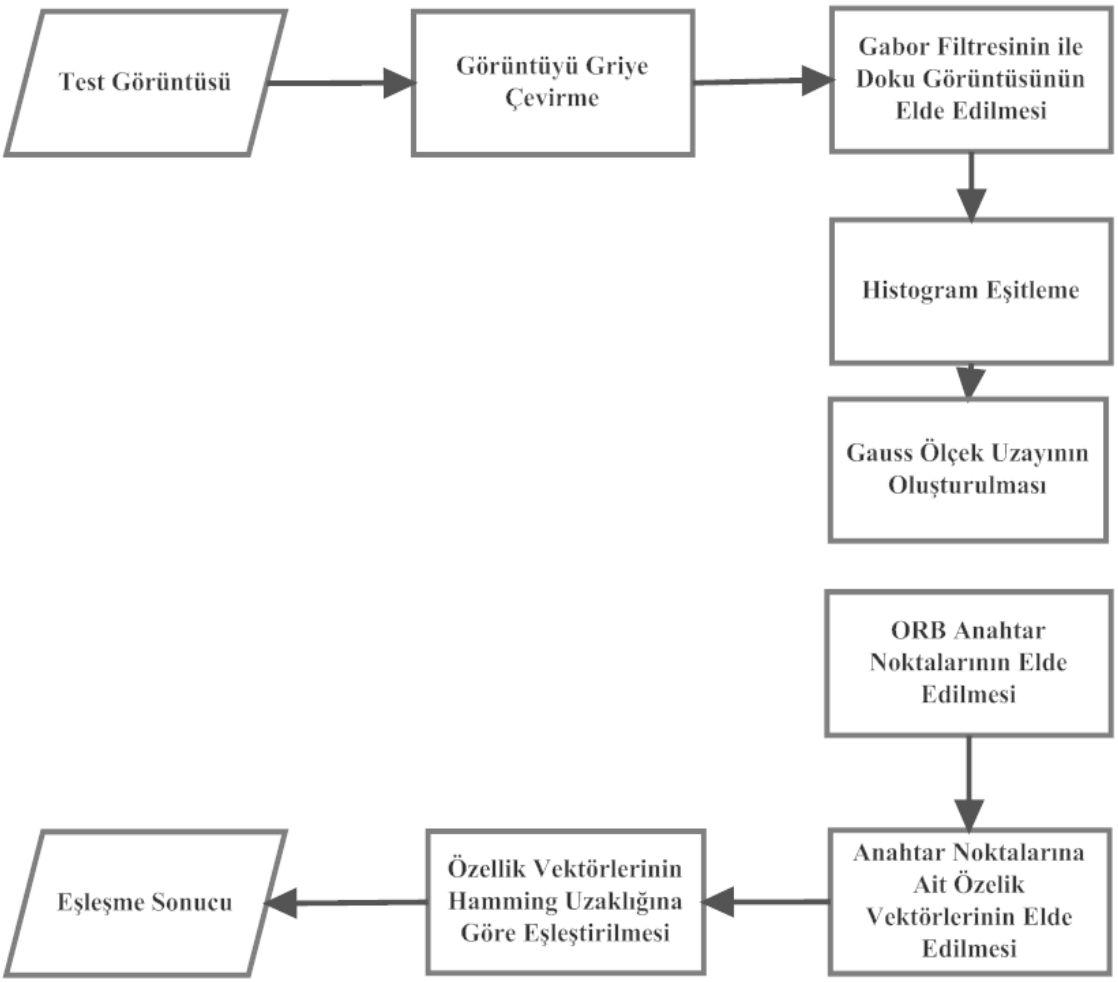


(a) (b) (c)

Şekil 2. 9. (a) Orijinal görüntü (b) Sahte görüntü (c) Önerilen yöntem sonucu

2.4. Gabor Filtresi ve ORB Tabanlı Kopyala Yapıştır Sahteciliği Tespiti Yöntemi

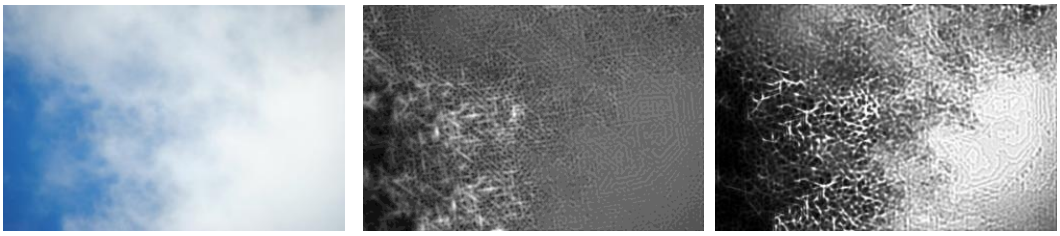
[38]'deki çalışmada ORB algoritmasının kullanılması ile görüntüden anahtar noktalarının elde edilmesi işlemi gerçekleştirilir. Anahtar noktası elde etme işleminde FAST köşe bulma algoritması kullanılmaktadır [42]. Ancak köşe bölgesi içermeyen özellikle düz (smooth) bölgelerde FAST algoritması anahtar noktası bulamayacağından etkin bir kopyala yapıştır sahteciliği tespiti yapılamayacaktır. Bu problemin üstesinden gelebilmek için tez kapsamında yapılan çalışmada görüntünün doku bilgisinin elde edilmesinin ardından ORB algoritmasının uygulanmasıyla daha fazla anahtar noktası elde edilerek kopyalanıp yapıştırılan bölgeye dair daha net bir sahtecilik tespiti yapılmıştır. Önerilen yöntemde gri seviyeye çevrilen görüntüden önce Gabor Filtresi ile doku görüntüsü elde edilmekte ve görüntünün netliği için histogram eşitleme uygulanmaktadır. Daha sonra ORB algoritması ile anahtar noktası tespit edilir ve bu anahtar noktalarına ait özellik vektörleri elde edilir. Elde edilen özellik vektörlerinin eşleştirilmesiyle kopyalanıp yapıştırılan bölgelerdeki eşleşme sonucu verilir. Önerilen bu yöntemin blok diyagramı Şekil 2.10'daki gibidir.



Şekil 2.10. Gabor filtresi ve ORB tabanlı kopyala yapıştır sahteciliği tespiti blok diyagramı

2.4.1. Gabor Doku Çıkarma ve Histogram Eşitleme

Gri seviyeye çevrilen test görüntüsünden doku çıkarma işlemini gerçekleştirmek için popüler olarak bilinen Gabor filtresi kullanılmaktadır. Ancak özellikle düz bölgelerde filtrelenmiş görüntü düşük kontrasta sahip olabileceğinden, görüntünün aynı piksel değerlerinin gri tonlarının daha geniş bir ölçek üzerine dağıtılmasını sağlayabilmek amacıyla histogram eşitleme (histogram equalization) [53] işlemi gerçekleştirilmiştir.



(a) (b) (c)

Şekil 2.11. (a) Sahte görüntü (b) Gabor filtresi sonucu (c) Histogram eşitleme sonucu

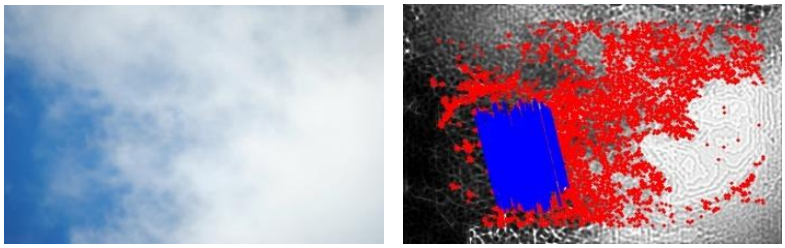
2.4.2. ORB Anahtar Noktalarının Elde Edilmesi ve Eşleştirilmesi

Bölüm 1.4.3' de de anlatıldığı gibi ORB algoritmasında anahtar noktalarının tespiti FAST köşe noktası bulunma algoritması kullanılarak gerçekleştirilir. Bu algoritma sayesinde anahtar noktaları hızlı bir şekilde elde edilir ancak görüntüde köşe noktası olarak tespit edilemeyecek düz bölgeler olduğunda ORB algoritması anahtar noktası çıkarmada başarısız olmaktadır. Dolayısıyla [38]'de önerilen yöntem özellikle düz bölgeler için doğru kopyala yapıştır sahteciliği tespiti yapamayacaktır. Bu yöntemin düz bölgelerde de kopyalanıp yapıştırılan bölgelerin etkin bir şekilde tespit edilebilmesi için görüntüden önce doku bilgisinin elde edilmesi ve ardından anahtar noktası çıkarma aşamalarının gerçekleştirilmesi yaklaşımı uygulanmıştır. Böylece doku görüntüsü üzerinde FAST (FAST-9) algoritması anahtar noktaların tespitini daha rahat yapabileceğinden [38]'deki çalışmaya göre daha etkin bir kopyala yapıştır sahteciliği önerilmiştir.

Histogram eşitlemesiyle birlikte daha net bir doku görüntüsünün elde edilmesinin ardından ORB algoritması kullanılarak bütün görüntüye ait anahtar noktalarının çıkarılması gerçekleştirilir. Şekil 2.12 (a)' da görüldüğü gibi ORB yöntemi kullanılarak yapılan anahtar noktası çıkarma işlemi ilgili bölgede hiç anahtar noktası bulamaz iken yapılan önişlemler sonrasında *7941* adet anahtar noktası çıkarılmıştır.

Anahtar noktalarına ait dönme bilgisi de hesaplandıktan sonra dönmeden bağımsız anahtar noktalarına ait *256* bitlik ikili (binary) özellik tanımlayıcı vektörleri elde edilmektedir. Daha sonra elde edilen bu ikili özellik vektörlerinin Hamming uzaklığına göre birbirine en yakın olan vektörlerin eşleştirilmesi işlemi gerçekleştirilmektedir.

Şekil 2.12 (a)'daki sahte görüntüyü ilişkin [38]' de önerilen yöntem ile hiç eşleşme bulunamamıştır ancak önerilen ön işlemler sayesinde Şekil 2.12 (b)'de görüldüğü gibi *296* tane eşleşme bulunmuştur.



(a) (b)

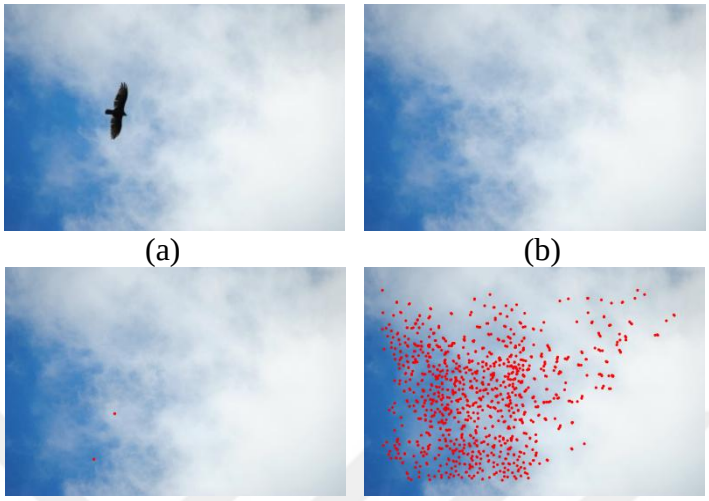
Şekil 2.12. (a) ORB[38] sonucu (b) Önerilen yöntemin eşleşme sonucu (Anahtar nokta sayısı:7941 Eşleşme sayısı:296)

2.5. AKAZE Tabanlı Kopyala Yapıştır Sahteciliği Tespiti

Literatürde anahtar noktası tabanlı kopyala yapıştır sahteciliği tespiti yöntemlerinde anahtar noktalarının çıkarımı ve özellik vektörlerinin elde edilmesi için SIFT [40] , SURF [41] ve ORB [42] algoritmaları kullanılmaktadır [31-39]. SIFT ve SURF algoritmalarının ikisinde de anahtar noktası çıkarma işleminden önce Gauss ölçek uzayı oluşturulmaktadır. Bu metotların en büyük dezavantajı nesne sınırlarını koruyamamasıdır. ORB'de ise anahtar noktalarının tespiti için FAST köşe bulma algoritması kullanılmasından dolayı köşe bilgisi içermeyen düz (smooth) bölgelerde anahtar noktası tespit edilememektedir. Dolayısıyla kopyala yapıştır sahteciliği tespitinde bu algoritmaların tek başına uygulanması durumunda etkin bir kopyala yapıştır sahteciliği tespiti yapılamayacaktır. Tez kapsamında yapılan ikinci ve üçüncü çalışmada görüntülerden doku bilgisi çıkarılarak bu problemin üstesinden gelinmeye çalışılmıştır. Ancak doku bilgisi elde etmek gibi ek bir ön işlem hesaplama süresinin uzamasına sebep olmaktadır. Tez süresince yapılan araştırmalarda düz bölgelerde anahtar noktası tespiti için doku bilgisi çıkarma ön işlemine ihtiyaç duymayan bir anahtar noktası çıkarma yöntemi araştırılmıştır. Literatür araştırmaları kapsamında Alcantarilla ve arkadaşlarının bu problemin üstesinden gelebilmek için 2012 yılında önermiş olduğu KAZE özellik çıkarma algoritması ve daha sonra yine aynı yazarlar tarafından önerilen AKAZE algoritması analiz edilmiştir [49,50]. [49]' da önerilen yöntemde kullanılan doğrusal olmayan filtreleme sayesinde ilgili çalışma SIFT ve SURF' e göre tekrar edilebilirliği ve ayırt edilebilirliği artırmıştır. AKAZE algoritması ise Bölüm 1.4.4' de de anlatıldığı gibi kullandığı FED yapısı sayesinde KAZE'ye göre daha hızlı çalışmaktadır.

AKAZE algoritmasının analizi ve uygulaması ile birlikte görüntülerden elde edilen anahtar nokta sayısının artışı gözlemlenmiştir. Buna dair bir örnek Şekil 2.13' de verilmiştir. SIFT algoritması ile sahte görüntü üzerinde *2* anahtar noktası elde edilirken SURF ve ORB algoritmaları ile (b) görüntüsünden anahtar noktası elde edilememektedir.

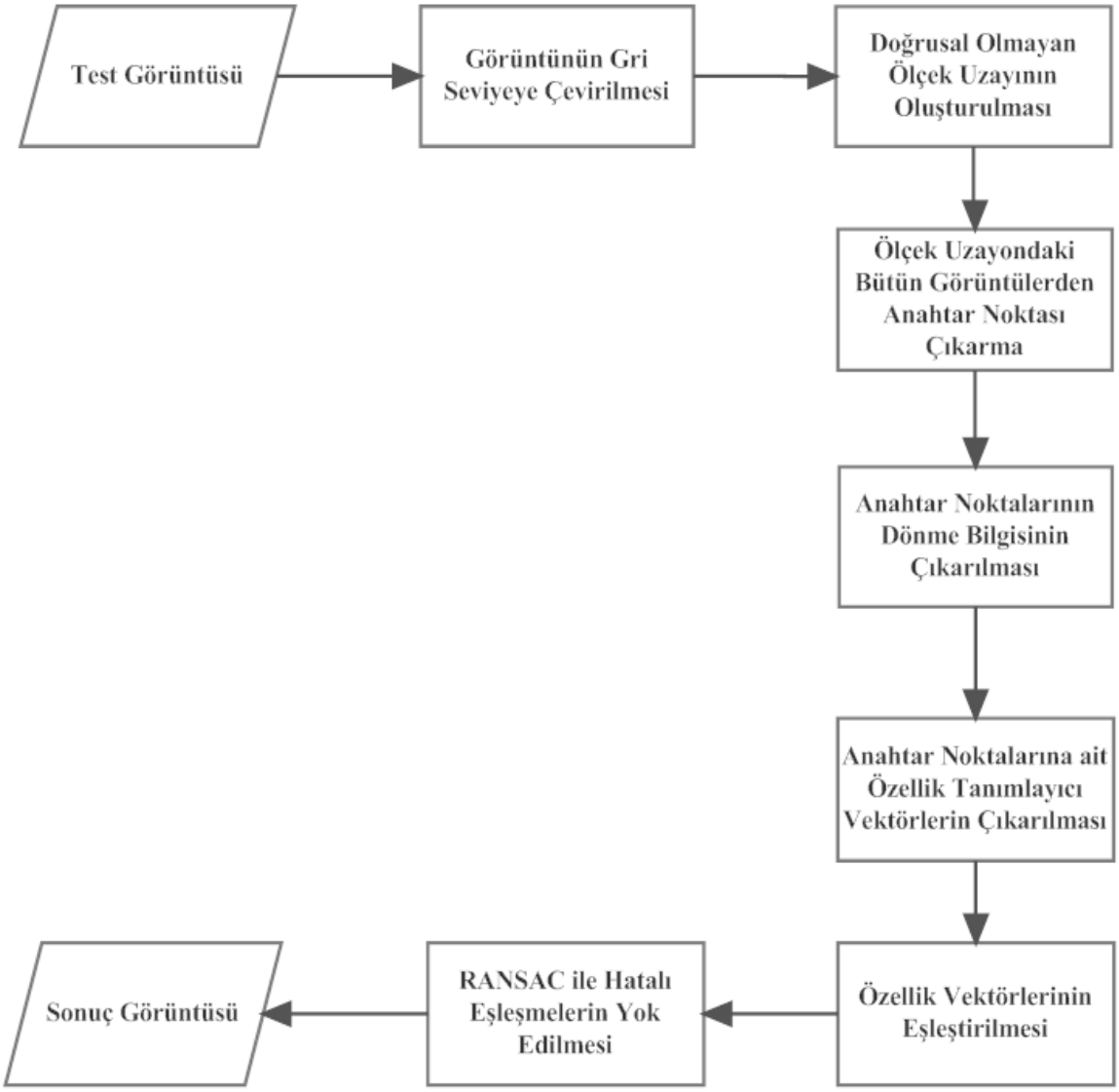
AKAZE algoritması ile ise *1210* adet anahtar noktası tespit edilmiştir. Şekil 2.13 (c)' de bu anahtar noktaları gösterilmiştir.



(a) (b) (c) (d)

Şekil 2.13. (a) Orijinal görüntü (b) Sahte görüntü (c) SIFT ile elde edilen anahtar noktalar (d) AKAZE ile elde edilen anahtar noktalar

AKAZE'nin anahtar noktası tespitindeki başarısından yola çıkarak AKAZE tabanlı yeni bir kopyala yapıştır tespiti yöntemi önerilmiştir. Önerilen yöntemde oluşturulan doğrusal olmayan ölçek uzayındaki görüntülerden anahtar noktaları ve bu anahtar noktalarına ilişkin özellik tanımlayıcı vektörleri elde edilmiştir. Elde edilen özellik vektörlerinin eşleştirilmesi işleminden sonra da varsa hatalı eşleştirmelerin yok edilmesi işlemi RANSAC algoritması yardımıyla gerçekleştirilmiştir. Önerilen yönteme ait blok diyagram Şekil 2.14' de verilmiştir.

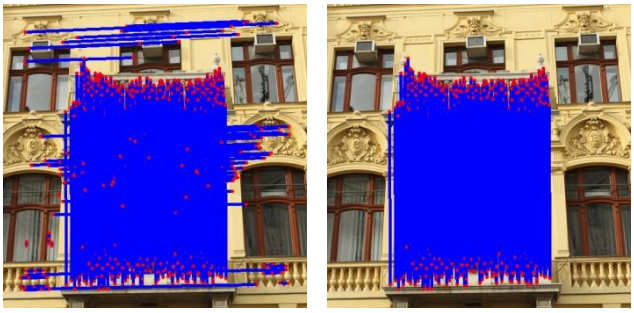


Şekil 2. 14. AKAZE Tabanlı kopyala yapıştır sahteciliği tespiti blok diyagramı

Önerilen yöntemde gri seviyeye çevrilen görüntüden dört ölçek ve dört oktav içeren doğrusal olmayan ölçek uzayı oluşturulmuştur. Doğrusal olmayan ölçek uzayının oluşturulmasında Bölüm 1.4.4.2' deki adımlar gerçekleştirilmiştir. Oluşturulan doğrusal olmayan ölçek uzayındaki her görüntü için bir anahtar noktası çıkarma işlemi gerçekleştirilir. Daha sonra elde edilen anahtar noktalarına ait özellik vektörlerinin çıkarılması işlemi gerçekleştirilmektedir. AKAZE anahtar noktalarına ait özellik tanımlayıcısını elde etmek için, LDB algoritmasını dönmeye karşı bağımsız bir sonuç üretecek şekilde modifiye edilmiş hali olan M-LDB algoritması kullanılmıştır [50]. Bunun için de öncelikle anahtar noktalarına ait dönme bilgisi SURF [42] algoritmasına benzer şekilde çıkartılmıştır. Anahtar noktalarına ait dönme bilgisi de elde edildikten sonra Bölüm

1.4.4.4'deki diğer işlem adımları da uygulanarak anahtar noktalarına ait *1x486* boyutunda ikilik özellik tanımlayıcı vektörler elde edilmiş olur. Bir sonraki aşamada ise elde edilen bu ikilik vektörlerin eşleştirilmesi işlemi gerçekleştirilmektedir. k adet anahtar noktasına ilişkin $A = A_1 \cdots A_k$ özellik tanımlayıcı vektörleri için her vektörün diğer bütün vektörlerle arasındaki uzaklığın tespiti için Hamming uzaklığı kullanılmıştır. Buna göre iki vektörün ikili elemanları arasındaki farklı değerler *XOR* fonksiyonu ile hesaplanmaktadır. Önceden belirlenen bir δ eşik değerine göre bu değerden küçük olan vektörlere sahip anahtar noktaları eşleştirilir ve eşleşen matrislerin koordinat bilgileri *M* eşleşme matrisinde tutulur. Yapılan çalışmada 486 boyutlu vektörlerin eşleştirilmesi için δ eşik değeri *35* olarak alınmıştır.

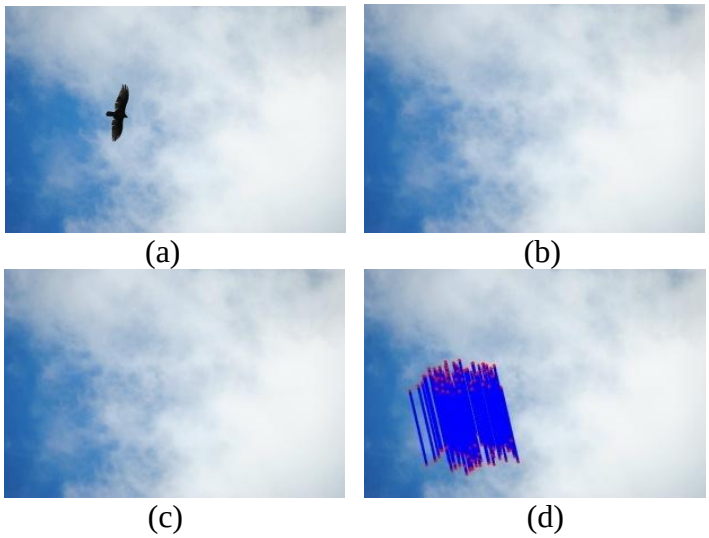
Eşleşme matrisi oluşturulduktan sonra var olan hatalı eşleşmelerin yok edilmesi için Bölüm 1. 7' de anlatılan RANSAC algoritması kullanılmıştır. Yapılan çalışmada RANSAC için rastgele eşleşen *5* anahtar noktası üzerinden ve *H* transformasyon matrisi hesaplanmaktadır. Transformasyon matrisi parametreleri ile eşleşen anahtar noktaları arasındaki Öklid uzaklığı için eşik değeri γ*=2.5* olarak alınmıştır. Şekil 2.15' de önerilen yöntem ile kopyala yapıştır sahteciliği tespiti örneği gerçekleştirilmiş olup Şekil 2.15 (a)' da görülen yanlış eşleştirmeler RANSAC algoritması ile yok edilerek Şekil 2.15 (b)' deki sonuç elde edilmiştir.



(a) (b)

Şekil 2. 15. (a) RANSAC'dan önce eşleşme sonucu (b) RANSAC'dan sonrası eşleşme sonucu

Önerilen yöntemin özellikle düz bölgeler ile nesne kapama amacıyla yapılan kopyala yapıştır sahteciliklerinin tespitinde literatürdeki SIFT tabanlı [35], SURF tabanlı [34] ve ORB tabanlı [38]' deki çalışmalara göre üstünlüğünü gösteren bir örnek Şekil 2.16' da verilmiştir.



Şekil 2.16. (a) Orijinal görüntü (b) Ataksız sahte görüntü (c) SIFT [35], SURF [35], ORB [38] sonucu (Doğru eşleşme sayısı:0) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:53)

2.6. Performans Değerlendirme Metrikleri

Bu bölümde tez kapsamında yapılan çalışmaların performans analizinin yapılabilmesi için kullanılan performans değerlendime metrikleri incelenecektir. Yapılan ilk üç çalışma için Tespit Oranı metriği, yapılan son çalışmada ise ROC analizi ile sonuç değerlendirilmesi yapılmıştır.

2.6.1. Tespit Oranı Metriği

Anahtar noktası tabanlı kopyala yapıştır sahteciliği tespiti yönteminin etkinliğinin analizinin yapılması için Tespit Oranı (TO) metriği tarafımızca önerilmiştir. Bu metrik *NxM* boyutlu test görüntüsünün sahtecilik tespiti kapasitesini değerlendirmektedir. *TO* metriği kopyalanıp yapıştırılan bölgelerdeki eşleşen anahtar nokta sayısının o bölgelerdeki piksel sayısına oranıdır. Bu oranın *NM/100* ile çarpılmasıyla da bu metriğin görüntü boyutundan bağımsızlığı sağlanmıştır. K_F, kopyalanan ve yapıştırılan bölge içinde tespit edilen anahtar nokta sayısını, $|F|$ kopyalanan bölgedeki piksel sayısını belirtmek üzere bu oran Eşitlik (2. 1)'deki gibi hesaplanmaktadır. Yüksek *TO* değerleri daha doğru tespit sonuçlarını ifade etmektedir.

$$TO = \left(\frac{K_F}{|F|}\right)\frac{NM}{100} \quad (2.\ 1)$$

2.6.2. ROC Analizi (Receiver Operating Characteristic Analysis)

İstatistiksel karar teorisine dayanan ROC analizi yöntemi 1950' lerin başında teknik bilimlerde sinyal belirleme analizi için geliştirilmiştir. İlk olarak 2. Dünya savaşı sırasında radar görüntülerinin analizinde kullanılmıştır. Daha sonra 1960' ların başında tıpta tanı testlerinin değerlendirilmesinde ROC eğrilerinin kullanılabileceği fikri ortaya atılmıştır. 1960' ların sonlarında tıp alanında görüntüleme araçlarının değerlendirilmesinde ROC analizi kullanılmaya başlanmıştır [56].

ROC analizi duyarlılık ve özgüllük arasındaki ilişkiyi incelemektedir. ROC analizi için çizilen ROC Eğrisi iki sınıflı bir sınıflama sistemi içinde değişen kesim noktalarına göre elde edilen duyarlılığa (doğru pozitif oran) karşı 1-Özgüllüğün (1-Doğru negatif oran) çizildiği bir grafiktir [56]. ROC analizi için kullanılacak olan duyarlılık ve özgüllük değerlerinin nasıl hesaplandığı Tablo 1. 1 yardımıyla aşağıdaki şekilde gösterilmeye çalışılmıştır.

Tablo 1. 1. ROC Eğrisi için doğru atama tablosu

		Gerçek Durum	
		Pozitif	*Negatif*
Test Sonucu	*Pozitif*	Doğru Pozitif , a	Yanlış Pozitif, c
	Negatif	Yanlış Negatif, b	Doğru Negatif, d
	Toplam	a+b	c+d

Duyarlılık= $\frac{a}{a+b}$ (2. 2)

Özgüllük= $\frac{d}{c+d}$ (2. 3)

1-Özgüllük= $\frac{c}{c+d}$ (2. 4)

Duyarlılık (Doğru Pozitif Oran, DPO) :Gerçek durumun pozitif olduğu durumda test sonucunun pozitif olabilme olasılığıdır.

Özgüllük (Doğru Negatif Oran, DNO) : Gerçek durumun negatif olduğu durumda test sonucunun negatif olabilme olasılığıdır. 1-Özgüllük değeri ise Yanlış Pozitif Oranı (YPO) vermektedir.

Kopyala yapıştır sahteciliği tespiti için ROC eğrisi çiziminde kullanılan Doğru Pozitif Oran (Duyarlılık) ve Yanlış Pozitif Oran (1-Özgüllük) değerleri Eşitlik (2. 5) ve (2. 6)' daki gibi hesaplanmaktadır.

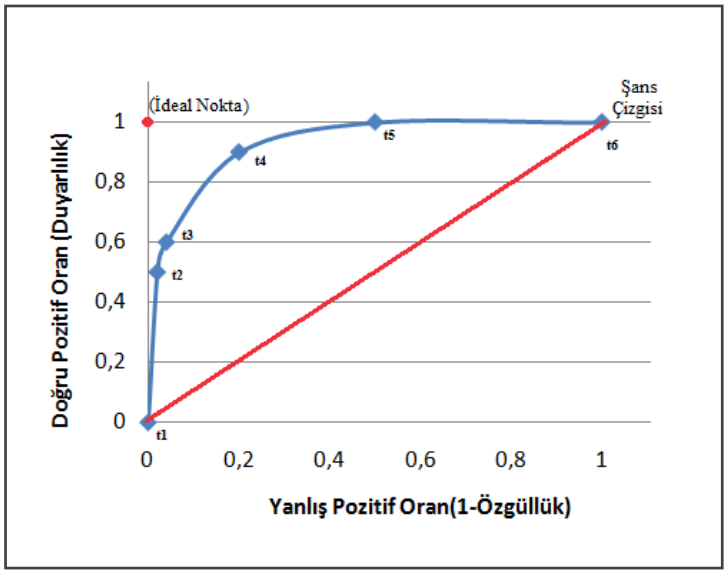
$$DPO = \frac{"Sahte\ Olarak\ \ Algılanan\ \ Görüntü\ Sayısı"}{"Sahte\ Görüntü\ Sayısı"} \tag{2.5}$$

$$YPO = \frac{"Orjinal\ \ Olduğu\ Halde\ \ Sahte\ Olarak\ \ Algılanan\ \ Görüntü\ Sayısı"}{"Orjinal\ \ Görüntü\ Sayısı"} \tag{2.6}$$

Tablo 1. 2'de örnek bir YPO ve DPO değerleri verilmiştir ve bu değerler ile çizilen ROC Eğrisi Şekil 2.17'deki gibidir.

Tablo 1. 2. ROC Eğrisi örneği için örnek YPO ve DPO değerleri

	t_1	t_2	t_3	t_4	t_5	t_6
YPO	0	0,02	0,04	0,2	0,5	1
DPO	0	0,5	0,6	0,9	1	1



Şekil 2.17. ROC Eğrisi örneği

2.6.3. Renkli SURF Tabanlı Kopyala yapıştır Sahteciliği Tespiti Deneysel Sonuçları

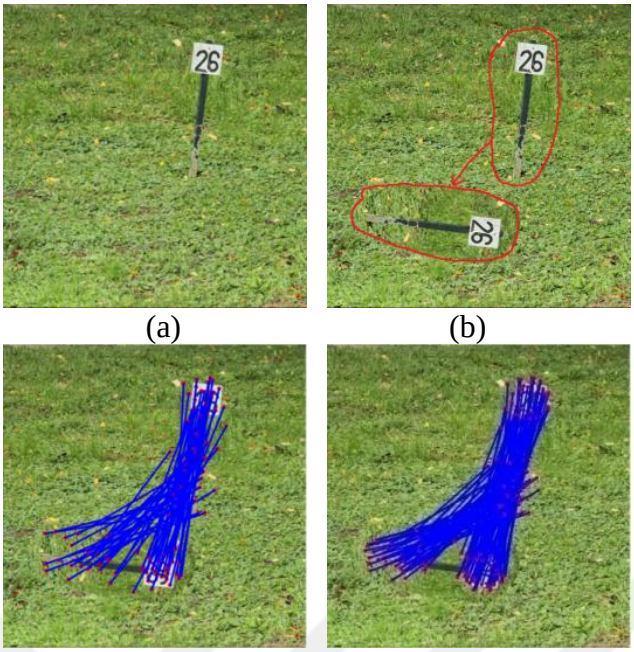
Tez kapsamında yapılan ilk çalışmanın oluşturulan veri seti üzerindeki performans sonuçları bu bölümde verilmiştir. Elde edilen sonuçlar literatürdeki SURF anahtar noktası tabanlı yöntemlerden [34]'deki çalışma ile karşılaştırılmıştır. Deneyler önerilen yöntemin i7 Core 2. 3 GHz işlemcili Windows 7 işletim sistemli dizüstü bilgisayarda Matlab R2015a yazılımında kodlanmasıyla gerçekleştirilmiştir.

Önerilen yöntemin performans analizi için Comofod veri tabanındaki *512x512* boyutlu görüntülerden faydalanılmıştır [58]. Veritabanındaki görüntülere, GIMP açık kaynak kodlu görüntü düzenleyici program yardımıyla ekstra sahtecilik uygulanarak *40* adet sahte görüntü oluşturulmuştur.

Oluşturulan bu veri seti üzerinde önerilen yöntemin kopyala yapıştır sahteciliği tespiti kapasitesini değerlendirmek için Tespit Oranı (TO) metriği kullanılmıştır. Elde edilen sonuçlara ilişkin karşılaştırmalı örnek görsel sonuçlar ve ortalama *TO* sonuçları verilmiştir.

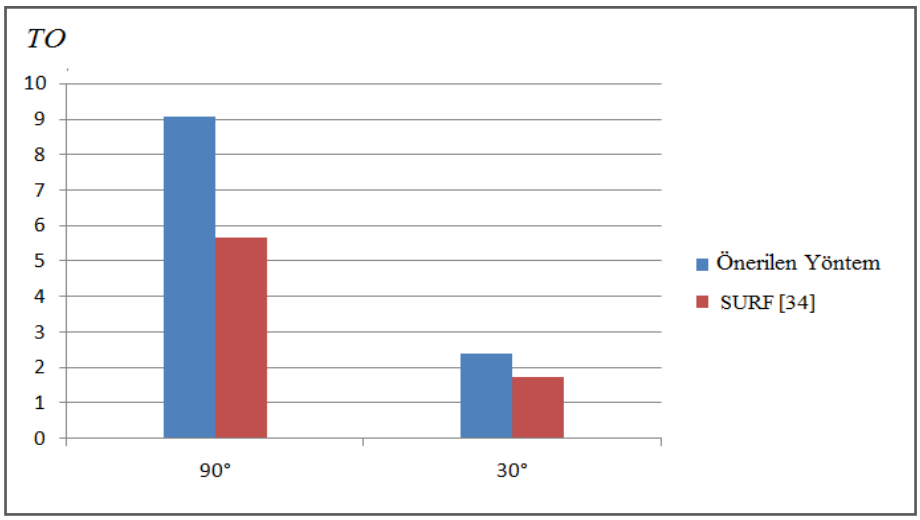
2.6.3.1. Dönme Atağı Altındaki Deneysel Sonuçlar

İlk olarak önerilen yöntemin kopyalanan bölgenin döndürülüp daha sonra yapıştırılması durumunda nasıl çalıştığı test edilmiştir. Buna göre veri tabanındaki *30* ve *90* derece döndürme atağına maruz kalmış *40* adet görüntü üzerinde test işlemi gerçekleştirilmiştir. Şekil 2.19'da önerilen yöntemin *90* derece dönme atağı uygulanmış sahte görüntü üzerinde performans sonucu verilmiştir. [34]'deki çalışma ve önerilen yöntem sonuçları Şekil 2.19 (c) ve (d)'de gösterilmiştir. Bu örnek için sırasıyla elde edilen eşleşme sayısı *26* ve *63'* tür.



Şekil. 2.19. (a) Orijinal görüntü (b) Sahte görüntü (c) [34]'de önerilen yöntem sonucu (Eşleşme sayısı: 26) (d)Önerilen yöntem sonucu (Eşleşme sayısı: 63)

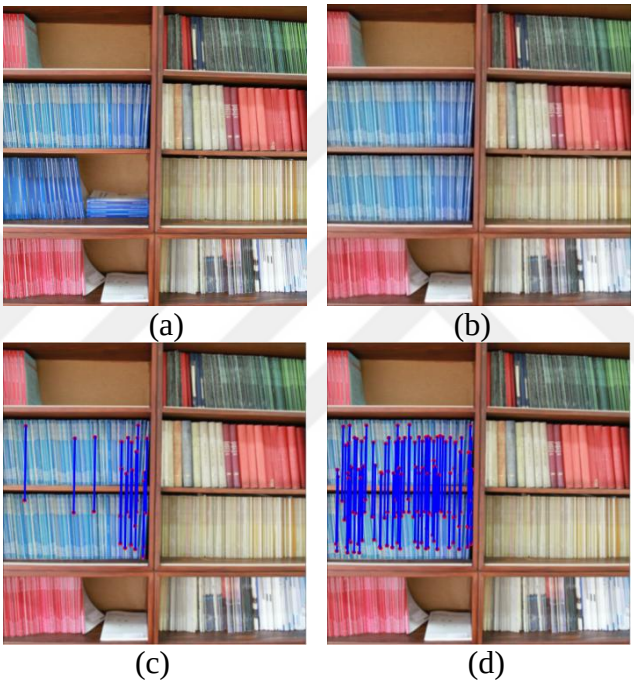
Önerilen yöntemin dönme atağı durumundaki performansına ilişkin ortalama bir sonuç elde etmek için bütün bu görüntülere [34]' deki yöntem ve önerilen yöntem uygulanarak ortalama *TO* değerleri elde edilmiştir. Şekil 2.20'de görüldüğü gibi belirtilen döndürme derecelerinde [34]'e göre daha başarılı sonuçlar elde edilmiştir.



Şekil 2. 20. Dönme atağı durumunda karşılaştırmalı test sonucu

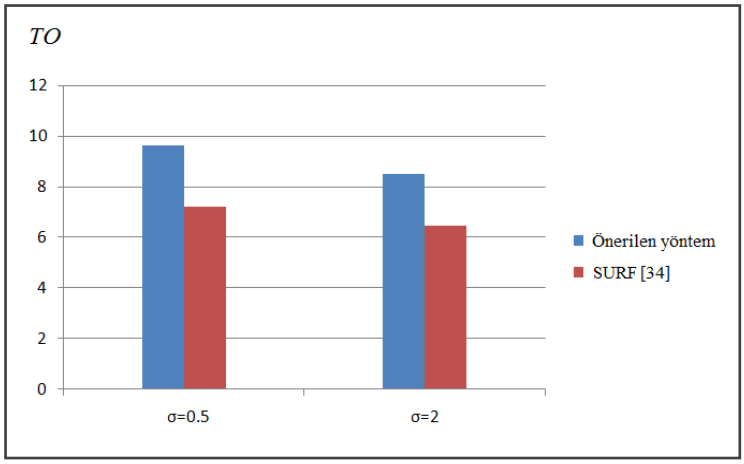
2.6.3.2. Bulanıklaştırma Atağı Altındaki Deneysel Sonuçlar

Bu bölümde önerilen yöntemin, Gauss bulanıklaştırma atağı durumunda performans analizi yapılmıştır. Şekil 2.21'de, pencere boyutu [*3x3*] ve σ değeri *2* olacak şekilde bulanıklaştırma atağına maruz kalmış bir örnek sahte görüntüye ait eşleşme sonuçları, [34] ve önerilen yöntem için verilmiştir. Şekil 2.21 (c) ve (d)'de sırasıyla eşleşme sonuçları görülmektedir. [34]' deki önerilen yöntem ile *16* adet eşleşme elde edilirken önerilen yöntem ile *40* adet eşleşme sonucu elde edilmiştir. Önerilen yöntemin SURF tabanlı [34]' deki çalışmaya göre daha yüksek doğruluğa sahip olduğu görülmektedir.



Şekil 2.21. (a) Orijinal görüntü (b) Sahte görüntü (c) [34]'deki yöntem sonucu (Eşleşme sayısı: 16) (d) Önerilen yöntem sonucu (Eşleşme sayısı: 40)

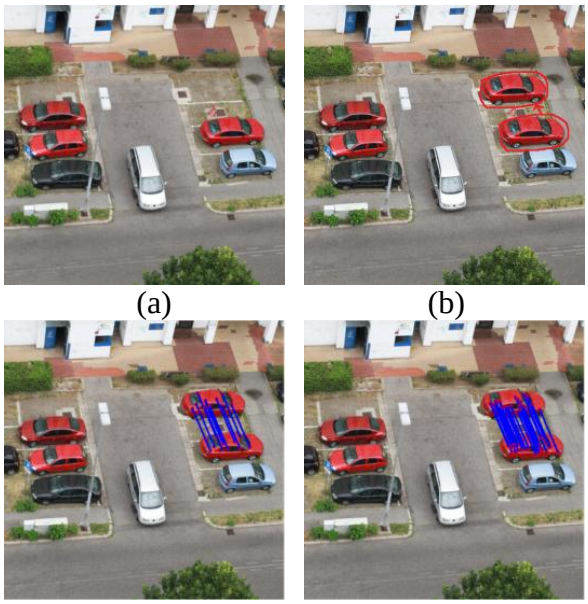
Ortalama bir sonuç elde etmek için veri setindeki 40 test görüntüsüne pencere boyutu [*3x3*] olacak şekilde σ*=0. 5* ve σ*=2* değerleri ile bulanıklaştırma uygulanmıştır. Şekil 2.22' de önerilen yöntemin, [34]' deki yönteme göre bu veri seti üzerinde daha yüksek ortalama tespit oranına sahip olduğu görülmektedir.



Şekil 2.22. Gauss bulanıklaştırma atağı durumunda karşılaştırmalı test sonucu

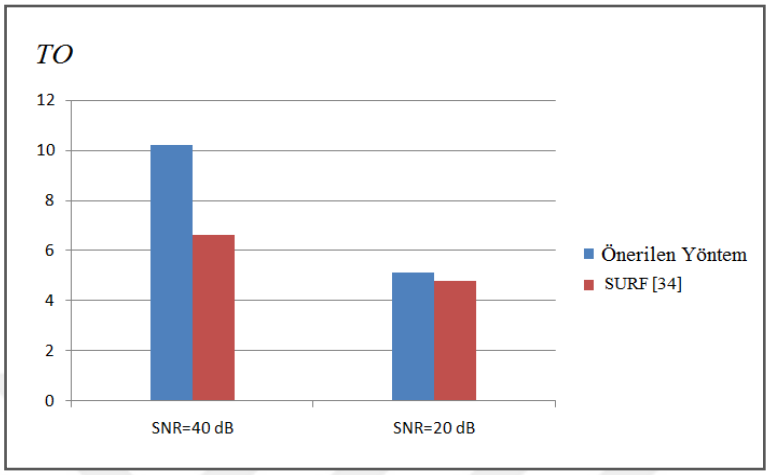
2.6.3.3. AWGN Atağı Altındaki Deneysel Sonuçlar

Son deneyde ise önerilen yöntemin AWGN atağına karşı dayanıklılığı gözlemlenmiştir. Şekil 2.23'deki örnekte (a)'daki orijinal görüntünün kopyala yapıştır işleminden sonra SNR değeri *20 dB* olacak şekilde AWGN atağına maruz kalması sonucu Şekil 2.23 (b) görüntüsü elde edilmiştir. Bu görüntüde, [34]'de önerilen yöntem ile *17* adet eşleşme tespiti yapılırken önerilen yöntem ile bu sayı *32* olmaktadır. Şekil 2.23 (c) ve 2.23(d)'de de [34] ve önerilen yöntem için görsel eşleşme sonuçları verilmiştir.



Şekil 2.23. (a) Orijinal görüntü (b) Sahte görüntü (c) [34]' deki sonucu (Eşleşme sayısı:17) (d) Önerilen yöntem (Eşleşme sayısı: 32)

Veri setindeki bütün görüntülere *20 dB* ve *40 dB* sinyalleri uygulanarak bu atağa karşı ortalama performans sonucu elde edilmiştir. Şekil 2.24'de de görüldüğü gibi bu atak durumunda bile önerilen yöntem [34]'e göre daha yüksek performans sergilemiştir.

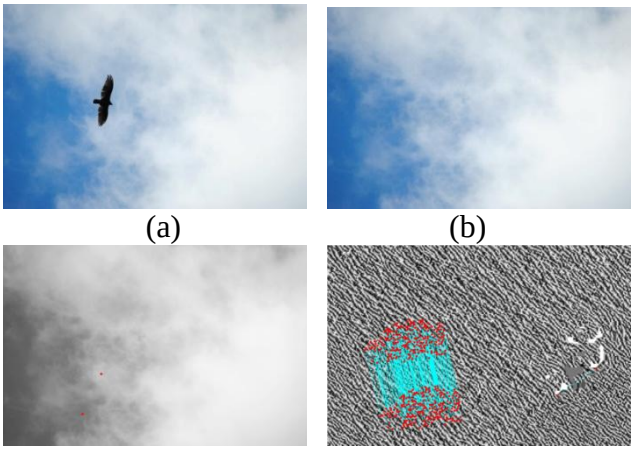


Şekil 2.24. AWGN atağı durumunda karşılaştırmalı test sonucu

2.6.4. LPQ ve SIFT Tabanlı Kopyala yapıştır Sahteciliği Tespiti Deneysel Sonuçları

Bu bölümde yapılan ikinci çalışma olan, LPQ ve SIFT tabanlı kopyala yapıştır sahteciliği tespiti uygulamasının performansına dair görsel sonuçlar ve ortalama performansa ilişkin deneysel sonuçlar verilmiştir. Bu deney için kullanılan veri seti Google görsel sonuçlardan elde edilen görüntülerin GIMP ile oluşturulan sahte hallerini içermektedir [57]. Önerilen yöntemin bulanıklaştırma, JPEG sıkıştırma ve AWGN atakları durumundaki performans analizini yapabilmek için veri setindeki görüntülere ayrıca bahsi geçen ataklar da uygulanmıştır. Yapılan çalışmanın performans değerlendirmesinde Tespit Oranı (TO) metriği kullanılmıştır. Elde edilen sonuçlara ilişkin karşılaştırmalı örnek görsel sonuçlar ve ortalama *TO* sonuçları verilmiştir.

Şekil 2.25 (a)'daki orijinal görüntüde yer alan kuşu içeren bölge tamamen kapatılmıştır. (c)'de de görüldüğü gibi bu sahte görüntü üzerinde [35]'de önerilen yöntem sadece *2* anahtar noktası bulmuş olup hiç eşleşme bulamamıştır. Önerilen yöntem ise 2.25 (d)'de görüldüğü gibi doku çıkarma işleminin sağladığı fayda ile *206* adet eşleşme bulmaktadır.

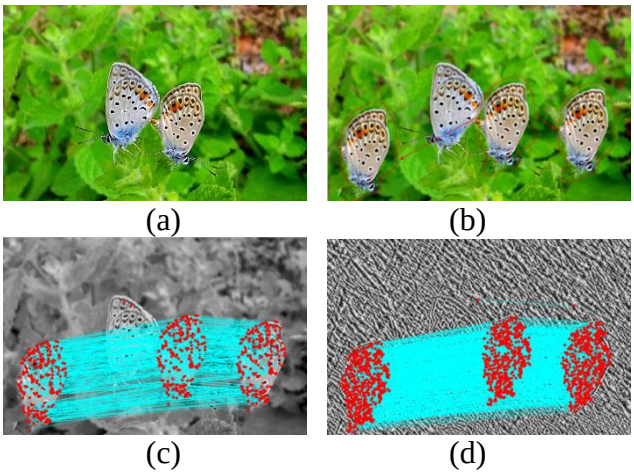


(a) (b)

(c) (d)

Şekil 2.25. (a) Orijinal görüntü (b) Sahte görüntü (c) [35]'de önerilen yöntem sonucu (Eşleşme sayısı:0) (d) Önerilen yöntem sonucu (Eşleşme sayısı: 206)

Yapılan çalışmanın çoklu kopyala yapıştır uygulanan görüntüler üzerinde performans analizini yapmak için Şekil 2.26 (a)'daki kelebeğin 2.26 (b)'deki gibi iki kere çoğaltılması gerçekleştirilmiştir. Şekil 2.26 (c)'de [35]' de önerilen yöntemin sonucunda *180* adet eşleşme bulunurken, bu sonuç önerilen yöntem ile *426'* ya çıkmaktadır. Böylece doku çıkarmanın sağladığı fayda karmaşık bölgelerde hatta çoklu kopyala yapıştır sahteciliğinde de görülmüş olur.



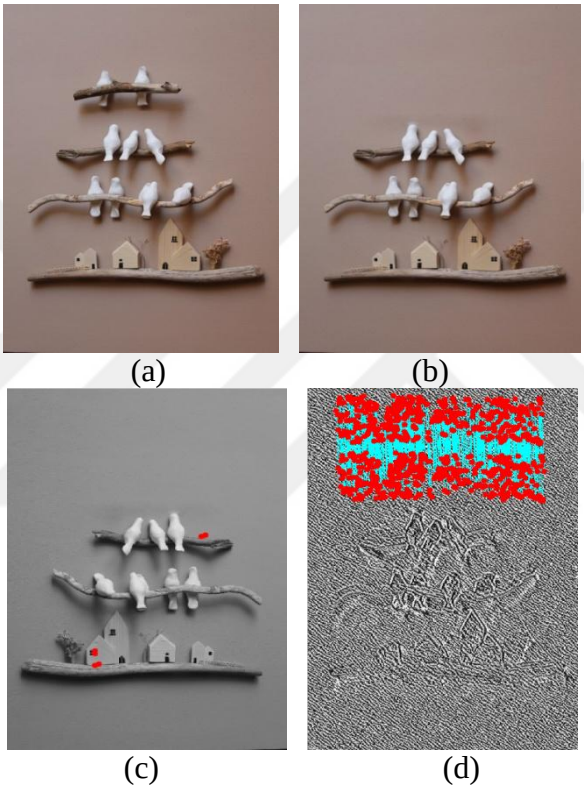
(a) (b)

(c) (d)

Şekil 2.26. (a) Orijinal görüntü (b) Sahte görüntü (c) [35]' de önerilen yöntem sonucu (Eşleşme sayısı:180) (d) Önerilen yöntem sonucu (Eşleşme sayısı:426)

2.6.4.1. JPEG Sıkıştırma Atağı Altındaki Deneysel Sonuçlar

Önerilen yöntemin JPEG sıkıştırma atağı gerçekleştirildiği durumlarda bile etkin çalışabildiğini gösterebilmek amacı ile test işlemleri gerçekleştirilmiştir. Şekil 2.27'de ilgili test işlemine ilişkin olarak kalite faktörü (KF) değeri *90* olacak şekilde sıkıştırılmış 2.27 (b)'deki sahte görüntünün [35]'deki yöntem ile önerilen yöntem sonucu verilmiştir. 2.27 (c)'de görüldüğü gibi kapatılan düz bölgede hiç eşleşme bulunamamıştır. Ancak 2.27 (d)'de verilen önerilen yöntem sonucunda görüldüğü gibi *393* eşleşme bulunmuştur.

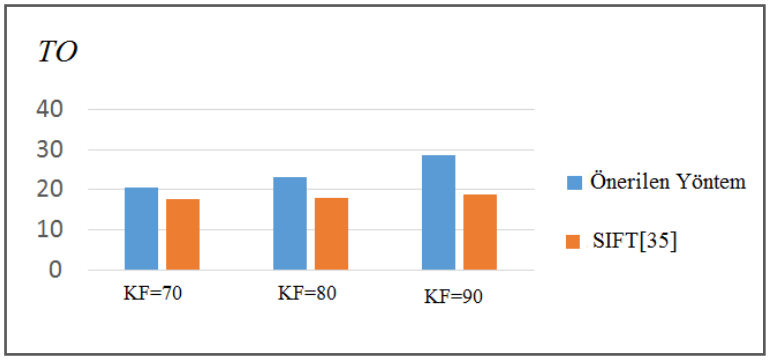


(a) (b)

(c) (d)

Şekil 2. 27. (a) Orijinal görüntü (b) KF=90 ile sıkıştırılmış sahte görüntü (c) [35]'de önerilen yöntem sonucu (Eşleşme sayısı:3) (d) Önerilen yöntem sonucu (Eşleşme sayısı:393)

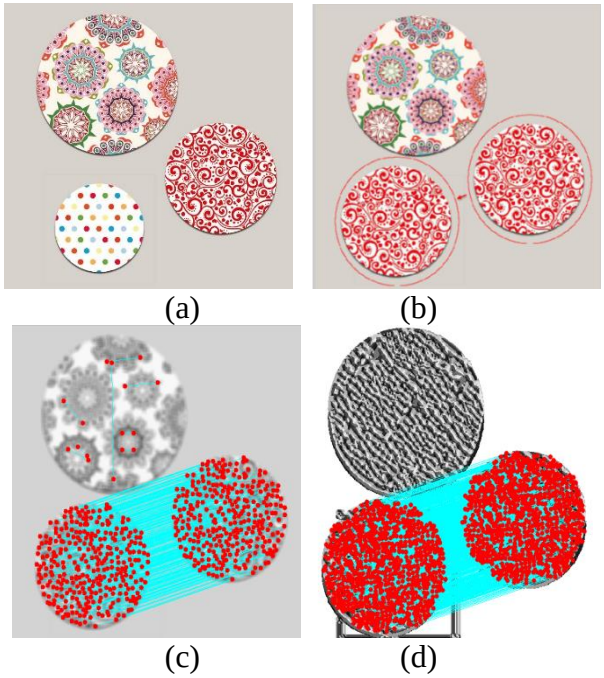
JPEG sıkıştırma atağında önerilen yöntemin [35] ile karşılaştırmalı performans analizini yapabilmek için *1200x800* boyutlu *40* adet görüntü *KF=90,80* ve *70* ile sıkıştırarak test görüntüleri oluşturulmuştur. Şekil 2.28'de de görüldüğü gibi bu üç durumda da önerilen yöntemin daha yüksek ortalama *TO* değerlerine sahip olduğu ortaya konulmuştur.



Şekil 2.28. JPEG sıkıştırma atağı durumunda karşılaştırmalı test sonucu

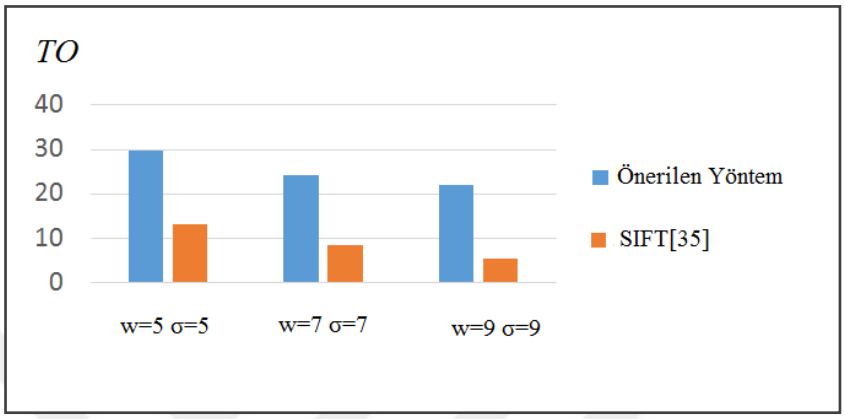
2.6.4.2. Gauss Bulanıklaştırma Atağı Altındaki Deneysel Sonuçlar

Bir sonraki test aşamasında, önerilen yöntemin Gauss atağı gerçekleştirildiği durumlarda bile etkin çalışabildiğinin gözlemlenmesi gerçekleştirilmiştir. Şekil 2.29' da ilgili test işlemine ilişkin, pencere boyutu $w=9$ ve $\sigma=9$ olacak şekilde Gauss bulanıklaştırılmış Şekil 2.29 (b)'deki sahte görüntünün [35]'deki yöntem ile önerilen yöntem sonucu verilmiştir. Şekil 2.29 (c)'de görüldüğü gibi SIFT [35] ile bulunan *318* adet eşleşme, doku çıkarma ön işlemiyle birlikte *894'* e çıkmıştır.



Şekil 2.29. (a) Orijinal görüntü (b) Sahte görüntü (c) [35]'de önerilen yöntem sonucu (Eşleşme sayısı:318) (d) Önerilen yöntem sonucu (Eşleşme sayısı:894)

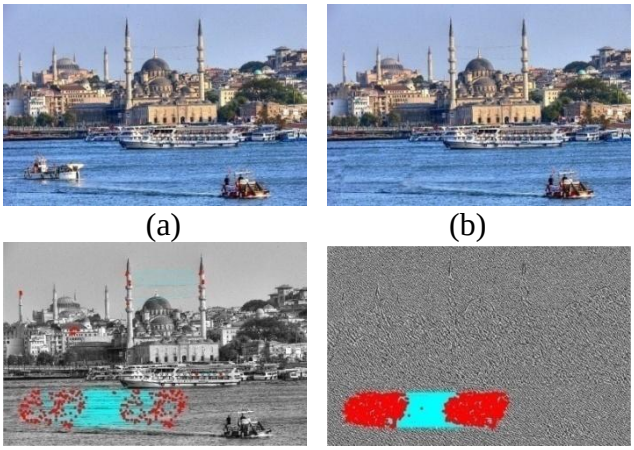
Gauss bulanıklaştırma atağında önerilen yöntemin [35]'deki çalışma ile ortalama performans analizini yapabilmek için *640 x 420* boyutlu 50 adet görüntü $\sigma = 5$, $\sigma = 7$ *ve* $\sigma = 9$ değerleri ve w=5, w=7, w=9 çerçeve boyutlarıyla bulanıklaştırılarak test görüntüleri oluşturulmuştur. Şekil 2.30'da da görüldüğü gibi bu üç durumda da önerilen yöntemin daha yüksek ortalama *TO* değerlerine sahip olduğu ortaya konulmuştur.



Şekil 2.30. Gauss bulanıklaştırma atağı durumunda karşılaştırmalı test sonucu

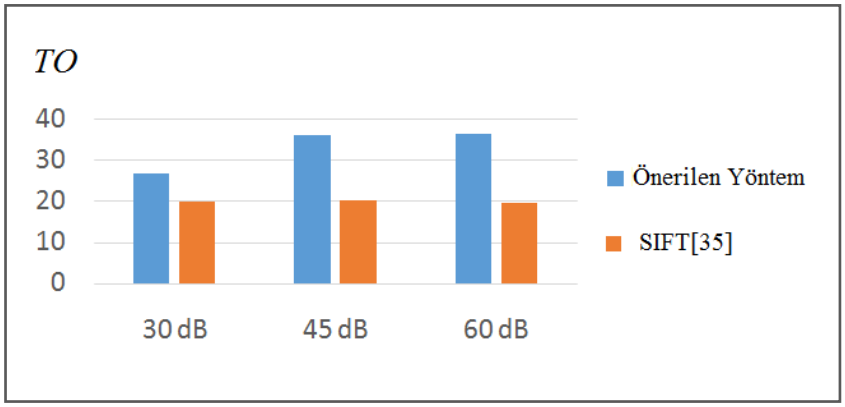
2.6.4.3. AWGN Atağı Altındaki Deneysel Sonuçlar

Son test işleminde ise Şekil 2.31 (a)'daki orijinal görüntüye ilişkin nesne kapama uygulaması gerçekleştirilmiş ve *60 dB* sinyalli AWGN gürültüsü eklenerek Şekil 2.31 (b) görüntüsü elde edilmiş ve bu görüntüye ilişkin görsel sonuç sunulmuştur. Şekil 2.31 (c)'de [35]'de önerilen yöntem kullanılarak kopyalanıp yapıştırılan bölgeler arasında *132* adet eşleşme bulmasına karşın önerilen yöntemde bu eşleşme sayısı *659'* a çıkmıştır.



Şekil 2.31. (a) Orijinal görüntü (b) Sahte görüntü (c) [35]'de önerilen yöntem sonucu (Eşleşme sayısı:132) (d) Önerilen yöntem sonucu (Eşleşme sayısı: 659)

AWGN (Additive White Gaussian Noise) atağı durumunda ortalama bir performans değerlendirmesi yapmak için 50 adet sahte görüntüye SNR değerleri *30 dB, 45 dB* ve *60 dB* olan sinyaller ile gürültü eklenmiştir. Şekil 2.32'de SIFT [35] ve önerilen yönteme ait ortalama TO değeri verilmiştir. Burada da görüldüğü gibi bu atak durumunda bile önerilen yöntem SIFT [35]'e göre daha yüksek performans sergilemektedir.

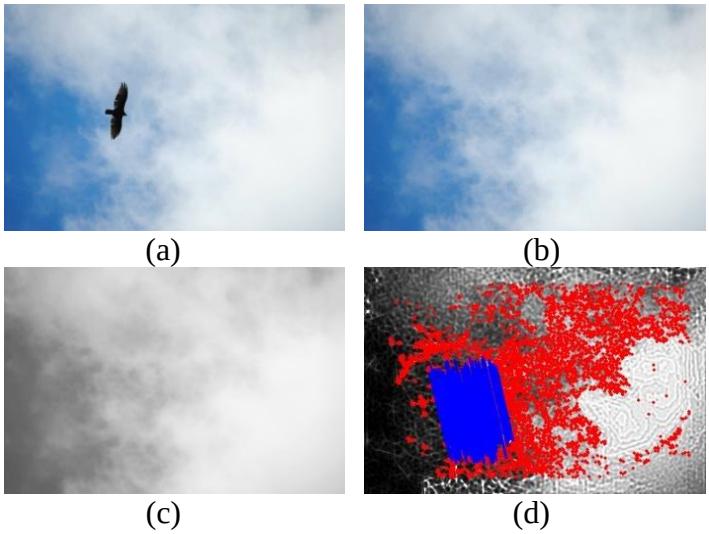


Şekil. 2.32. AWGN atağı durumunda karşılaştırmalı test sonucu

2.6.5. Gabor ve ORB Tabanlı Kopyala yapıştır Sahteciliği Tespiti Deneysel Sonuçları

Bu bölümde yapılan çalışmada Gabor filtresi ve ORB tabanlı kopyala yapıştır sahteciliği tespiti uygulamasının etkili performansına dair görsel sonuçlar ve ortalama performansa ilişkin deneysel sonuçlar verilmiştir. Bu deney için kullanılan veri seti Google görsel sonuçlardan elde edilen görüntülerin GIMP ile oluşturulan sahte hallerini içermektedir [55]. Önerilen yöntemin bulanıklaştırma ve JPEG sıkıştırma atakları durumundaki performans analizini yapabilmek için veri setindeki görüntülere ayrıca bu ataklar da uygulanmıştır. Yapılan bu çalışmanın performans değerlendirilmesinde Tespit Oranı(TO) metriği kullanılmıştır. Elde edilen sonuçlara ilişkin [38]' deki yöntem ile karşılaştırmalı örnek görsel sonuçlar ve ortalama *TO* sonuçları verilmiştir.

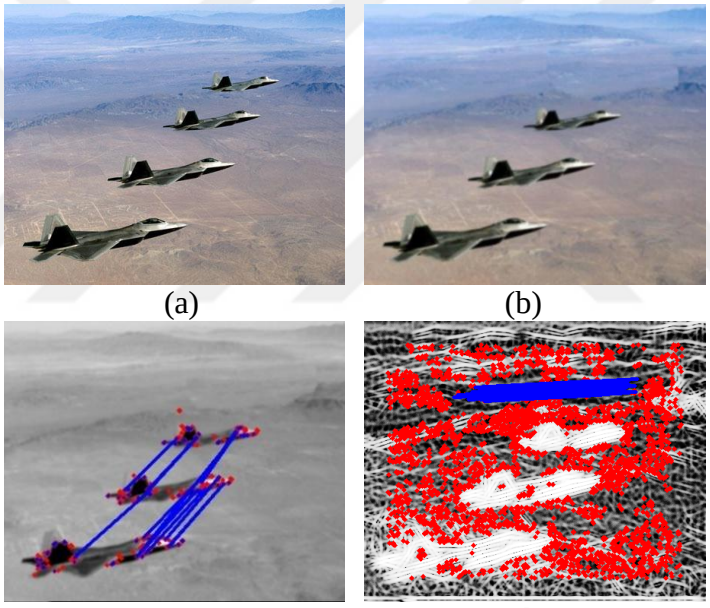
Şekil 2.33 (a)'daki orijinal görüntüdeki kuşu içeren bölge tamamen kapatılarak Şekil 2.33 (b)'deki ataksız sahte görüntü elde edilmiştir. Şekil 2.33 (c)'de de görüldüğü gibi bu sahte görüntü üzerinde [38]'de önerilen yöntem sadece *2* anahtar noktası bulmuş olup hiç eşleşme bulamamıştır. Önerilen yöntemde ise doku bilgisi elde edilen Şekil 2.33 (b) görüntüsünden *7941* adet anahtar noktası çıkarılarak *594* adet eşleşme bulunmuştur. Şekil 2.33 (d)'de eşleşme sonucu görülmektedir.



Şekil 2.33. (a) Orijinal görüntü (b) Sahte görüntü (c) ORB [38] sonucu (Doğru eşleşme sayısı:0) (d) Önerilen yöntem sonucu (Doğru eşleşme sayısı: 594)

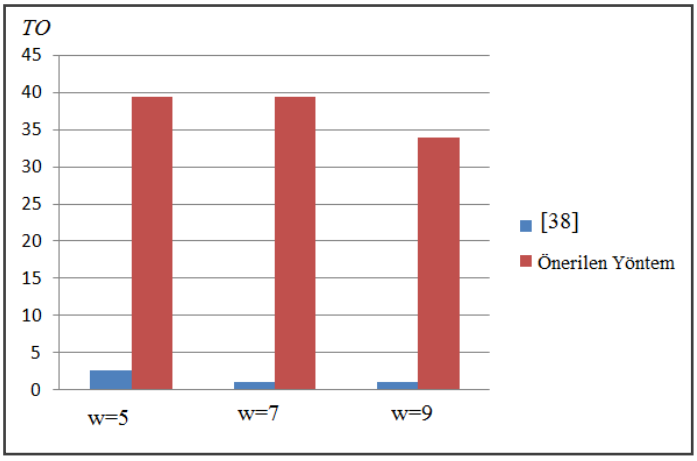
2.6.5.1. Gauss Bulanıklaştırma Atağı Altındaki Deneysel Sonuçlar

İlk olarak önerilen yöntemin Gauss bulanıklaştırma atağı gerçekleştirildiğindeki kopyala yapıştır sahteciliği tespiti performans analizi yapılmıştır. Şekil 2.34 (a)'daki orijinal görüntüdeki en üstte bulunan uçak düz gökyüzü ile tamamen kapatılarak (b)'deki sahte görüntü elde edilmiştir. Bu görüntü ayrıca $\sigma=5$ ve pencere boyutu $w=5$ olacak şekilde Gauss bulanıklaştırma atağına maruz bırakılmıştır. [38]'de önerilen yöntem bu görüntüden *360* adet anahtar noktası çıkarmasına karşın doğru eşleşme bulamamıştır. Şekil 2.34 (c)'de de görüldüğü gibi yanlış eşleşmeler mevcuttur. Önerilen yöntemde ise doku bilgisi elde edilen Şekil 2.34 (b) görüntüsünden *4418* adet anahtar noktası çıkarılarak *44* adet eşleşme bulunmuştur. Şekil 2.34 (d)'de eşleşme sonucu görülmektedir.



Şekil 2.34. (a) Orijinal görüntü (b) Sahte görüntü (c) ORB [38] sonucu (Doğru eşleşme sayısı:0) (d) Önerilen yöntem sonucu (Doğru eşleşme sayısı: 44)

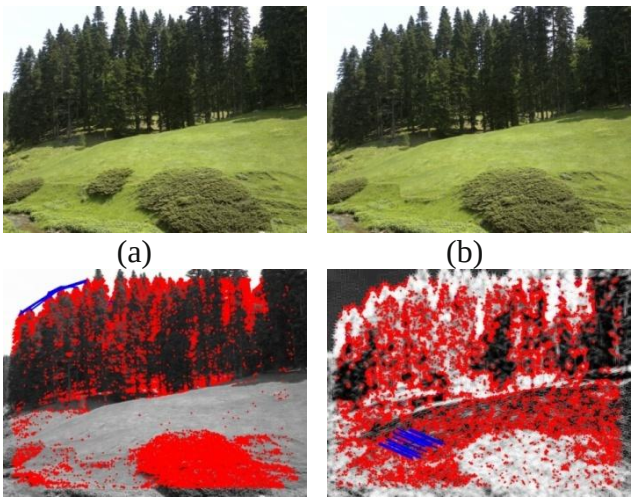
Gauss bulanıklaştırma atağında önerilen yöntemin [38] ile ortalama performans analizini yapabilmek için *40* adet test görüntüsü $\sigma = 5$ olmak üzere *5x5*, *7x7* ve *9x9* çerçeve boyutlarıyla bulanıklaştırılmıştır. Şekil 2.35' de de görüldüğü gibi bu üç durumda da önerilen yöntemin [38]'e göre daha yüksek ortalama *TO* değerlerine sahip olduğu ortaya konulmuştur.



Şekil 2.35. Gauss bulanıklaştırma atağı durumunda karşılaştırmalı test sonucu

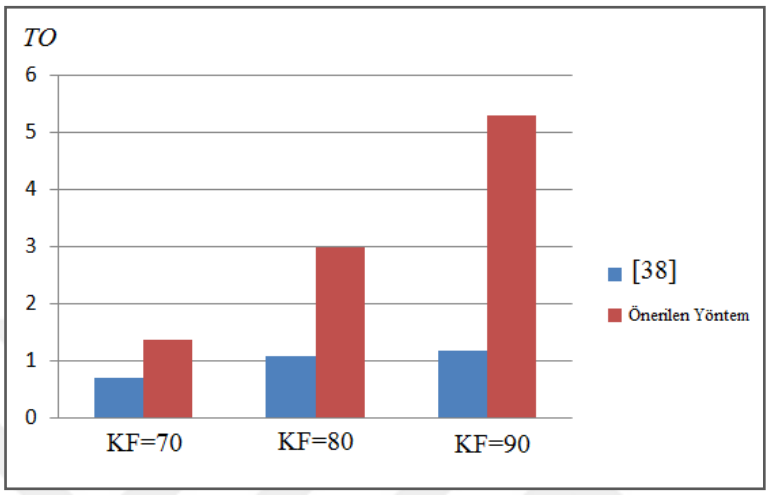
2.6.5.2. JPEG Sıkıştırma Atağı Altındaki Deneysel Sonuçlar

Önerilen yöntemin JPEG sıkıştırma atağı gerçekleştirildiği durumlarda bile etkin çalışabildiğini göstermek amacıyla çeşitli testler gerçekleştirilmiştir. Şekil 2.36 ' da ilgili test işlemine ilişkin kalite faktörü (KF) değeri *90* olacak şekilde sıkıştırılarak Şekil 2.36 (b)'deki sahte görüntünün, [38]'deki yöntem ve önerilen yöntem ile elde edilen sonucu verilmiştir. Şekil 2.36 (c)'de görüldüğü gibi düz bölge ile kapatılan bölgede doğru eşleşme bulunamamıştır. Ancak önerilen Şekil 2.36 (d)'de verilen yöntem sonucunda görüldüğü gibi *24* eşleşme bulunmuştur.



Şekil 2.36. (a) Orijinal görüntü (b) Sahte görüntü (c) ORB [38]'deki yöntem sonucu (Doğru eşleşme sayısı:0) (d) Önerilen yöntem sonucu (Doğru eşleşme sayısı: 24)

JPEG sıkıştırma atağında önerilen yöntemin [38] ile karşılaştırmalı performans analizini yapılabilmesi için *40* adet görüntü *KF=90*, *80* ve *70* ile sıkıştırılarak test görüntüleri oluşturulmuştur. Şekil 2.37'de de görüldüğü gibi bu üç durumda da önerilen yöntemin daha yüksek ortalama *TO* değerlerine sahip olduğu ortaya konulmuştur.



Şekil 2.37. JPEG sıkıştırma atağı durumunda karşılaştırmalı test sonucu

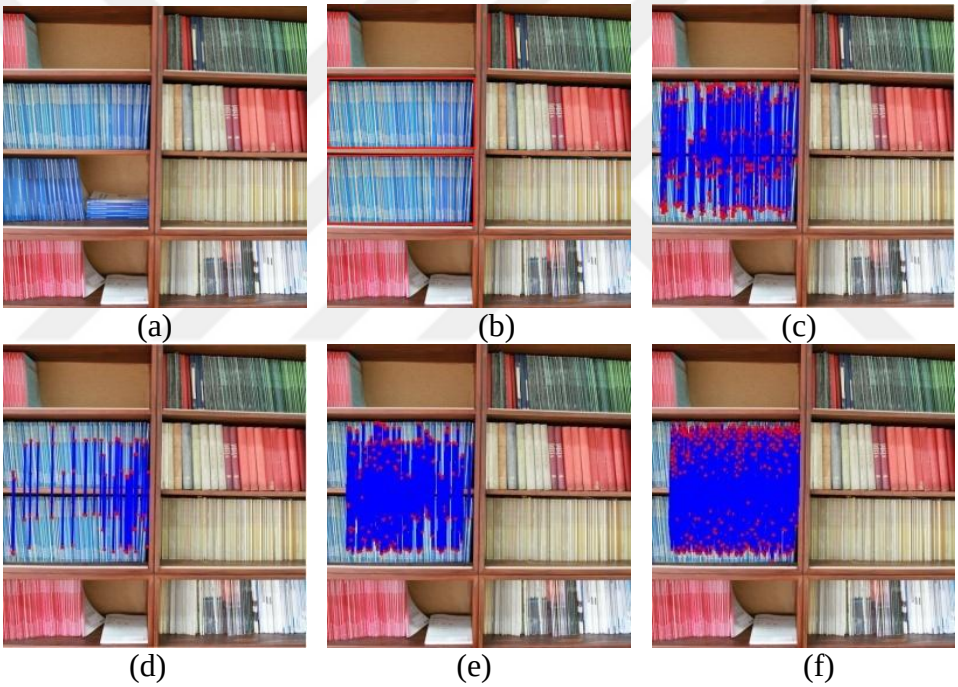
2.6.6. AKAZE Tabanlı Kopyala Yapıştır Sahteciliği Tespiti Deneysel Sonuçları

Tez kapsamında yapılan son çalışmanın oluşturulan veri seti üzerindeki performans sonuçları bu bölümde verilmiştir. Deneyler i7 Core 2.3 GHz işlemcili Windows 7 işletim sistemli dizüstü bilgisayarda OpenCV yazılımında kodlanmasıyla gerçekleştirilmiştir.

Önerilen yöntemin performans analizi için Google görsel sonuçlarından [57] ve Comofod veri tabanından [58] elde edilen 80 adet görüntüye GIMP açık kaynak kodlu görüntü düzenleyici programı yardımıyla sahte görüntüler oluşturulmuştur. Bu görüntülerin %40' ına nesne kapama amaçlı; %60'ına nesne veya bölge çoğaltma amaçlı kopyala yapıştır sahteciliği yapılmıştır. Ayrıca veri setindeki görüntülerin her biri dönme, JPEG sıkıştırma, Gauss bulanıklaştırma ve AWGN ataklarına maruz bırakılarak önerilen yöntemin bu ataklara karşı dayanıklılığı test edilmek istenilmiştir. Böylece veri tabanında 320 adet sahte görüntünün yer aldığı söylenebilir. Literatürdeki anahtar noktası tabanlı yöntemlerden SIFT tabanlı [35], SURF tabanlı [34] ve ORB tabanlı [38]'de önerilen yöntemler de bu veri seti üzerinde icrası gerçekleştirilmiş ve önerilen yöntem ile

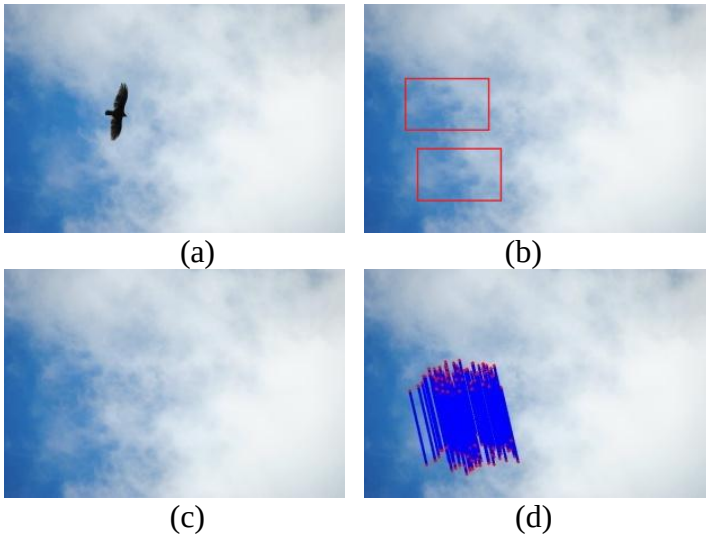
karşılaştırılması yapılmıştır. Karşılaştırmalı performans analizi için örnek görsel sonuçlar ve ilgili analizin ROC eğrileri verilmiştir.

Önerilen yöntemin temel başarısını görüntülemek için ataksız görüntülerde gerçekleştirilen iki örnek uygulama Şekil 2.38 ve Şekil 2.39' da verilmiştir. Şekil 2.38 (a)'daki görüntüye basit kopyala yapıştır sahteciliği uygulanarak Şekil 2.38 (b) görüntüsü elde edilmiştir. Bu görüntüdeki kopyala yapıştır sahteciliğinin tespiti için önerilen yöntemlerden SIFT tabanlı [35]' deki çalışma *110* tane, SURF tabanlı [34]' deki çalışma *41* ve ORB tabanlı [38]' deki çalışma *87* adet eşleşme bulmuştur. AKAZE tabanlı önerilen yöntem de ise eşleşme sayısı *324'e* çıkmaktadır. İlgili eşleşmeler sırasıyla Şekil 2.38 (c), (d), (e) ve (f)'de verilmiştir.



Şekil 2.38. (a) Orijinal görüntü (b) Ataksız sahte görüntü (c) SIFT [35] sonucu (Doğru eşleşme sayısı:110) (d) SURF [34] sonucu (Doğru eşleşme sayısı:41) (e) ORB [38] sonucu (Doğru eşleşme sayısı:87) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:324)

Bir sonraki deneyde, Şekil 2.39 (a)'daki kuşun (b)'deki gibi düz bir bölge ile kapatılan sahte görüntü üzerinde performans analizi yapılmıştır. Şekil 2.39 (b)'deki sahte görüntünün [35], [34] ve [38]' de önerilen yöntemlerin hiçbirinden doğru eşleşme sonucu alınamamıştır. Ancak önerilen yöntemde Şekil 2.39 (d)' deki gibi *53* adet eşleşme bulunarak görüntünün kopyala yapıştır sahteciliğine maruz kaldığı ortaya konulmuştur.

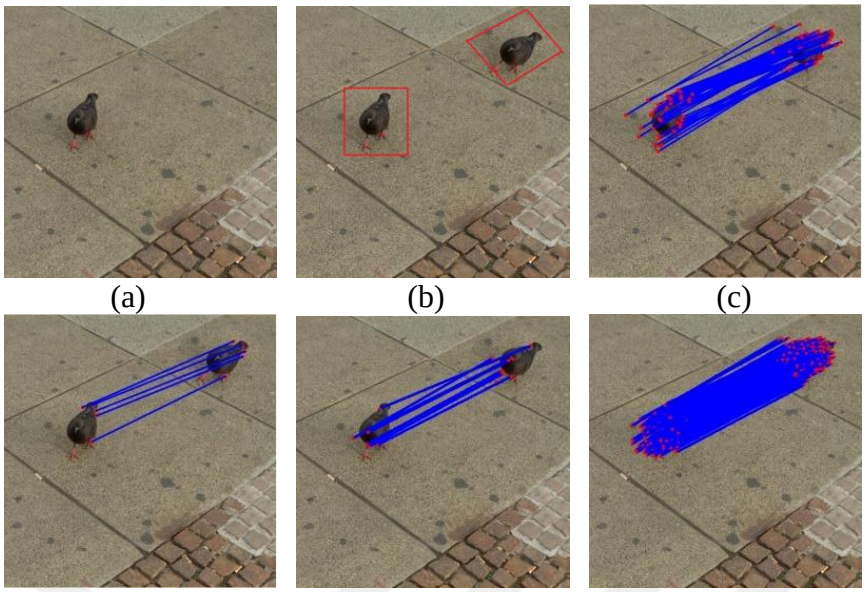


Şekil 2. 39. (a) Orijinal görüntü (b) Ataksız sahte görüntü (c) SIFT [35], SURF [34], ORB [38] sonucu (Doğru eşleşme sayısı:0) (d) Önerilen yöntem sonucu (Doğru eşleşme sayısı:53)

2.6.6.1. Dönme Atağı Altındaki Deneysel Sonuçlar

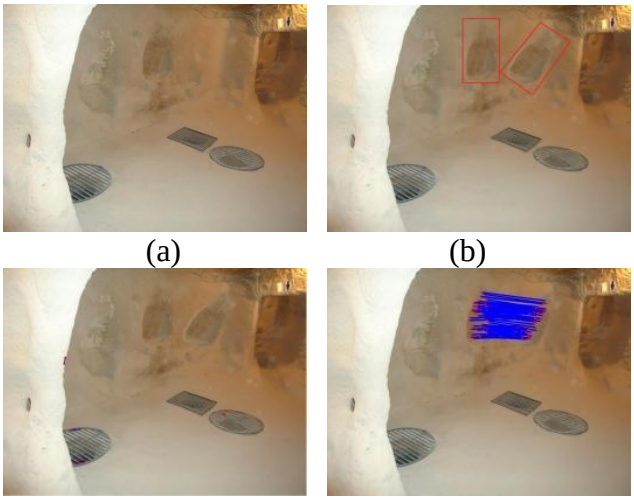
Önerilen yöntemin sahte görüntünün oluşturulması aşamasında dönme atağına maruz kalması durumunda da etkin bir şekilde çalışıp çalışmadığının analiz sonuçları bu bölümde verilmiştir. Test işlemi veri setindeki 30 ve 90 derecelik dönme atağına maruz kalan görüntüler üzerinde gerçekleştirilmiştir. Üç adet görsel sonuç ve veri setindeki bütün dönme ataklı görüntülere ilişkin elde edilen sonuçlarla çizilen ROC Eğrileri verilmiştir.

İlk olarak Şekil 2.40 (a)' daki görüntünün kopyalanan kuşku içeren bölgenin 30 derece döndürülüp yapıştırılmasıyla gerçekleştirilen sahteciliğin tespiti yapılmış ve görsel sonuçlar verilmiştir. SIFT tabanlı [35]' deki yöntem ile *28*, SURF tabanlı [34]'deki yöntem ile *5* ve ORB tabanlı [38]' deki yöntem ile *7* adet eşleşme bulunmuştur. Elde edilen görsel sonuçlar sırasıyla Şekil 2.40 (c), (d) ve (e)' de görülmektedir. AKAZE tabanlı önerilen yöntemde ise *95* adet eşleşme bulunarak bu performans üstünlüğü Şekil 2.40 (f)'de görülmektedir.



Şekil 2.40. (a) Orijinal görüntü (b) 30 derece dönme ataklı sahte görüntü (c) SIFT [35] sonucu (Doğru eşleşme sayısı:28) (d) SURF [34] (Doğru eşleşme sayısı:5) sonucu (e) ORB [38] sonucu (Doğru eşleşme sayısı:7)(f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:95)

Dönme atağının bir önceki örneğe göre daha düz bir bölgenin kopyalanıp sonra 30 derece döndürülüp yapıştırılmasıyla Şekil 2.41 (a)' daki orijinal görüntüden 2.41 (b)' deki sahte görüntü oluşturulmuştur. Bu sahte görüntüye uygulanan [35], [35] ve [38]' de önerilen yöntemler eşleşme bulamazken önerilen yöntem *37* adet eşleşme bularak görüntünün kopyala yapıştır sahteciliği içerdiği ortaya konulmuştur. Eşleşme sonucu Şekil 2.41 (d)'de verilmiştir.

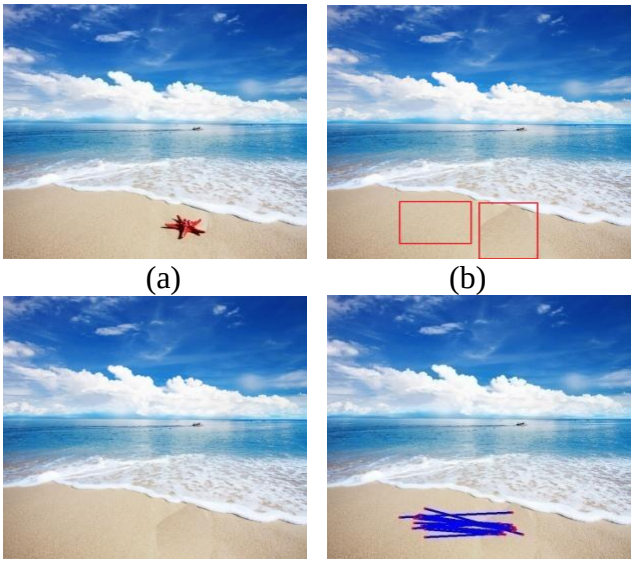


(a) (b)

(c) (d)

Şekil 2.41. (a) Orijinal görüntü (b) 30 derece dönme ataklı sahte görüntü (c) SIFT [35], SURF [34], ORB [38] sonucu (Doğru eşleşme sayısı:0) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:37)

Bir sonraki dönme atağı test işleminde ise Şekil 2.42 (a)' daki orijinal görüntüye 90 derecelik dönme atağı uygulanarak Şekil 2.42 (b)' deki sahte görüntü elde edilmiştir. Bu sahte görüntüye uygulanan [35], [34] ve [38]' de önerilen yöntemler eşleşme bulamazken önerilen yöntem *7* adet eşleşme bularak görüntünün kopyala yapıştır sahteciliği içerdiğini ortaya konulmuştur. Düz bölgelerde *90* derece dönme atağına maruz kalmış görüntüde bile sahtecilik tespitinin yapılabildiği ve örneğe ilişkin eşleşme sonucu Şekil 2.42 (d)' de verilmiştir.

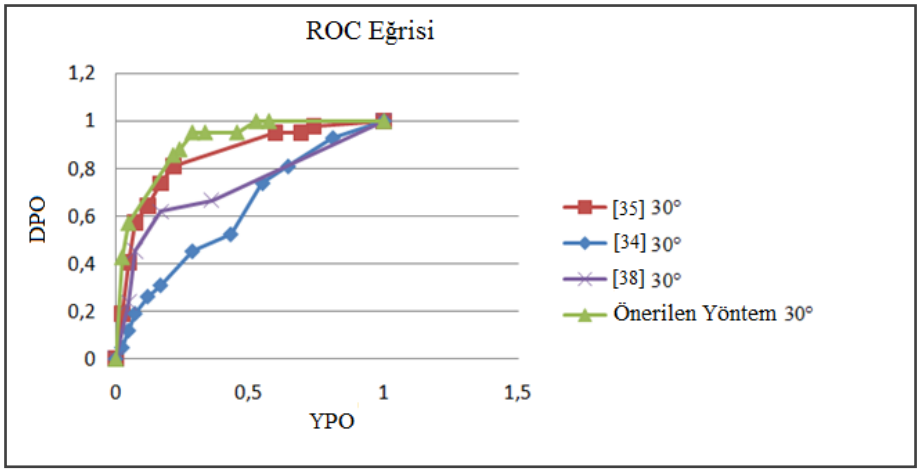


(a) (b)

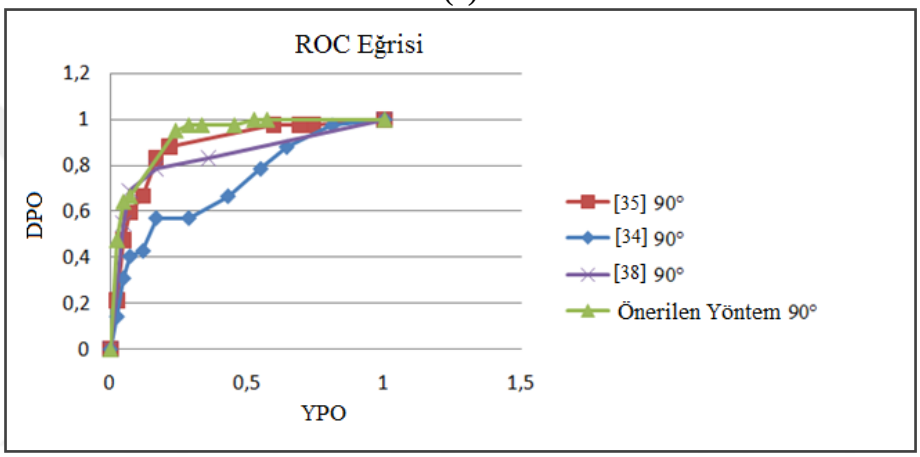
(c) (d)

Şekil 2.42. (a) Orijinal görüntü (b) 90 derece dönme ataklı sahte görüntü (c) SIFT [35], SURF [34], ORB [38] sonucu (Doğru eşleşme sayısı:0) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:7)

Genel bir performans analizi için veri setindeki dönme ataklı görüntülere SIFT tabanlı [35]' deki yöntem, SURF tabanlı [34]'deki yöntem, ORB tabanlı [38]' deki yöntem ve AKAZE tabalı önerilen yöntem uygulanmıştır. Şekil 2.43 (a)' da 30 derece dönme atağına maruz kalmış sahte görüntülerden, Şekil 2.43 (b)' de ise 90 derece dönme atağına maruz kalmış görüntülerden elde edilen sonuçlarla çizilen ROC eğrileri gösterilmiştir. ROC eğrilerinde, yeşil renkte çizilen eğri AKAZE tabanlı önerilen yöntemin deneysel sonuçlarından elde edilen DPO ve YPO değerlerine göre oluşturulmuştur ve diğer eğrilerde sırasıyla SIFT tabanlı [35], SURF tabanlı [34] ve ORB tabanlı [38]' deki çalışmaların sonuçlarına göre çizilmiştir. Görüldüğü gibi ideal nokta olan 1'e en yakın olan eğri, önerilen yöntem sonucunda elde edilen eğri olmuştur. Böylece önerilen yöntemin diğer yöntemlere göre daha iyi performans sergilediği söylenebilir.



(a)

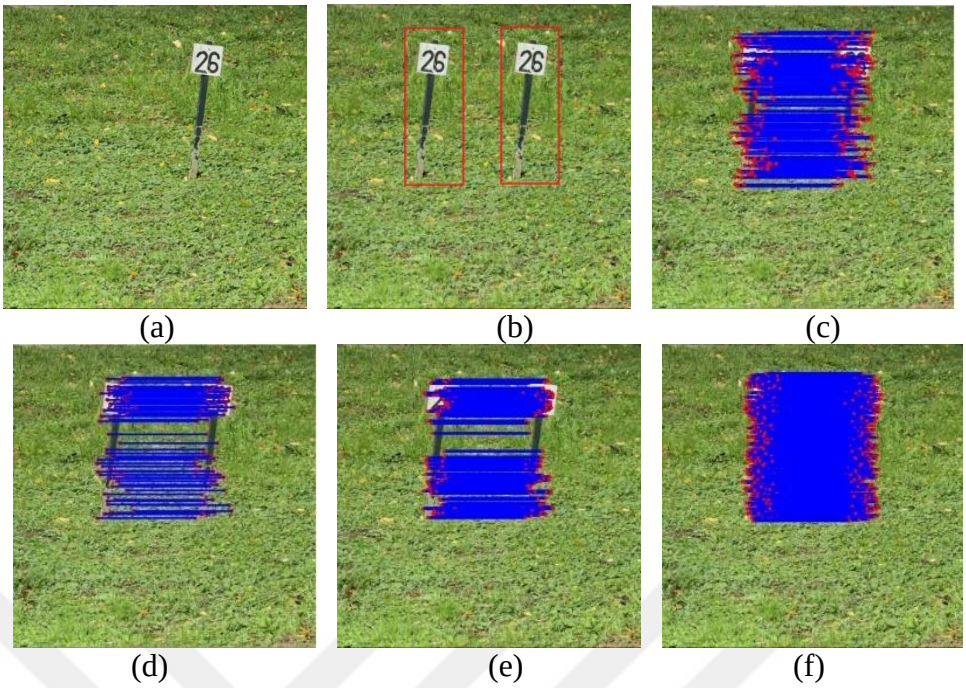


(b)

Şekil 2.43. (a) 30 derece dönme atağı durumunda (b) 90 derece dönme atağı durumunda SIFT [35], SURF [34], ORB [38] ve önerilen yöntemin ROC eğrileri

2.6.6.2. JPEG Sıkıştırma Atağı Altındaki Deneysel Sonuçlar

Bir sonraki deneyde Şekil 2.44 (a)'daki görüntüdeki bayrak bölgesinin çoğaltılarak sol tarafa yapıştırılmasının ardından *70* kalite faktörüyle sıkıştırılarak kaydedilmesiyle Şekil 2.44 (b)' deki sahte görüntü elde edilmiştir. SIFT tabanlı [35]'deki yöntem ile *109,* SURF tabanlı [34]' deki yöntem ile *74* ve ORB tabanlı [38]'deki yöntem ile *99* adet eşleşme bulunmuştur. Elde edilen görsel sonuçlar sırasıyla Şekil 2.44 (c), (d) ve (e)'de görülmektedir. AKAZE tabanlı önerilen yöntemde ise *192* adet eşleşme bulunarak performans üstünlüğü Şekil 2. 44 (f)'de gösterilmiştir.

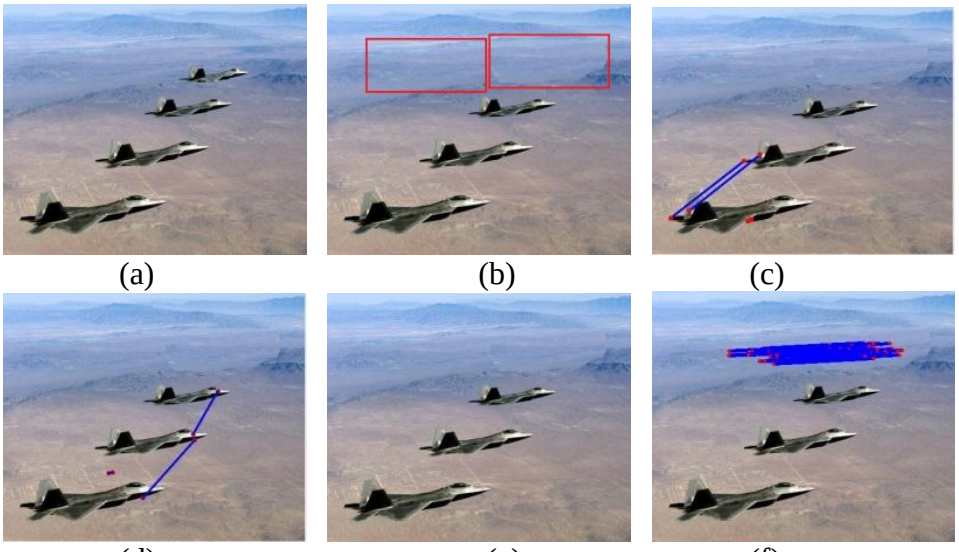


(a) (b) (c)

(d) (e) (f)

Şekil 2.44. (a) Orijinal görüntü (b) KF=70 ile JPEG sıkıştırma ataklı sahte görüntü (c) SIFT [35] sonucu (Eşleşme sayısı:109) (d) SURF [34] sonucu (Eşleşme sayısı:74) (e) ORB [38] sonucu (Eşleşme sayısı:99) (f) Önerilen yöntem sonucu (Eşleşme sayısı:192)

JPEG ataklı sahte görüntülerin tespitine ilişkin bir sonraki örnekte Şekil 2.45 (a)' daki uçak bölgesi düz gökyüzü bölgesi ile kapatılıp *90* kalite faktörü ile sıkıştırılmıştır. Elde edilen Şekil 2.45 (b)' deki verilen sahte görüntüye [35], [34] ve [38]'de önerilen yöntemler uygulanmıştır. Ancak bu yöntemlerin hiçbirinde doğru eşleşme bulunamamıştır. Önerilen yöntemin uygulanması ile birlikte ise *15* adet eşleşme bulunarak görüntünün kopyala yapıştır sahteciliğine maruz kaldığı ortaya konulmuştur. Önerilen yöntemin eşleşme sonucu Şekil 2.45 (f)'de görülmektedir.

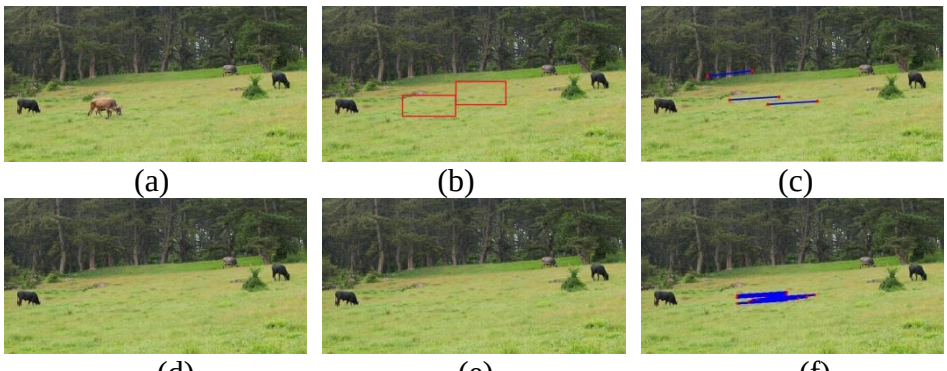


(a) (b) (c)

(d) (e) (f)

Şekil 2.45. (a) Orijinal görüntü (b) KF=90 ile JPEG sıkıştırma ataklı sahte görüntü (c) SIFT [35] sonucu (Doğru eşleşme sayısı:0) (d) SURF [34] sonucu (Doğru eşleşme sayısı:0) (e) ORB [38] sonucu (Doğru eşleşme sayısı:0) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:15)

Bir sonraki uygulamada Şekil 2.46 (a)' daki görüntüden elde edilen sahte görüntü 70 kalite faktörüyle sıkıştırılmıştır. SIFT [35] ile *2* doğru eşleşme tespit edilirken SURF [34] ve ORB [38]'den doğru eşleşme bulunamamıştır. Önerilen yöntem ise Şekil 2.46 (f)'de görüldüğü gibi *8* adet eşleşme tespit etmiştir.



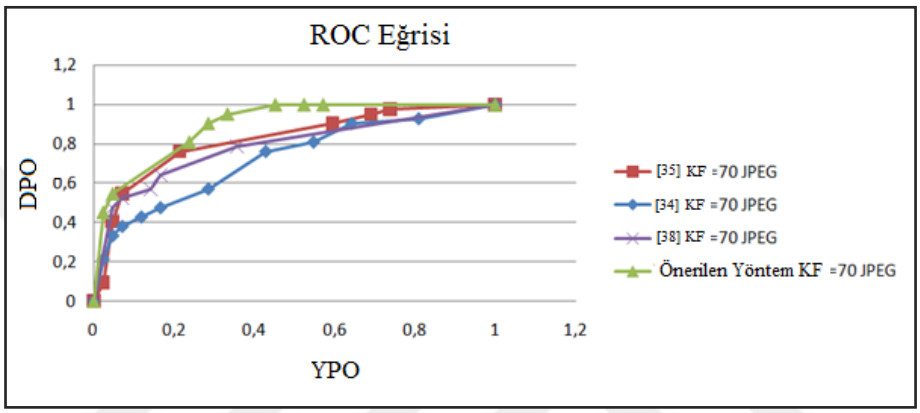
(a) (b) (c)

(d) (e) (f)

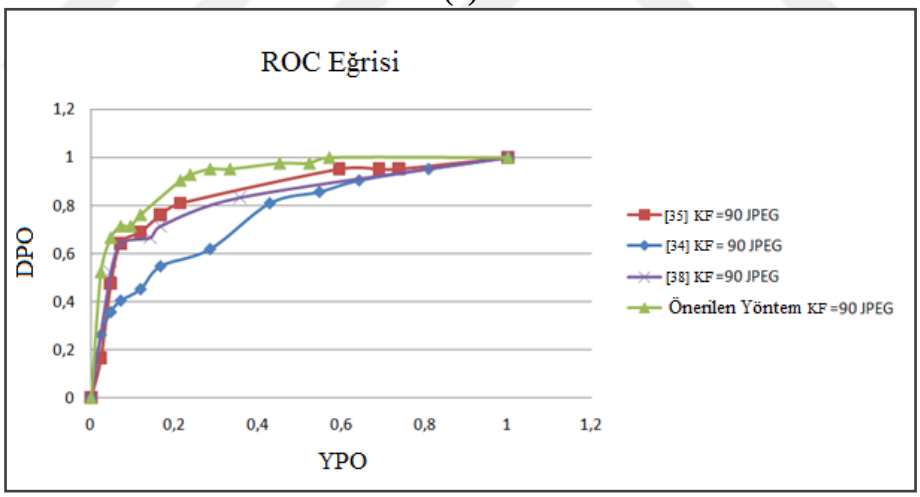
Şekil 2.46. (a) Orijinal görüntü (b) KF=70 ile JPEG sıkıştırma ataklı sahte görüntü (c) SIFT [35] sonucu (Doğru eşleşme sayısı:2) (d) SURF [34] sonucu (Doğru eşleşme sayısı:0) (e) ORB [38] sonucu(Doğru eşleşme sayısı:0) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:8)

JPEG sıkıştırma atağına karşı yöntemlerin gösterdiği performansın analizi için veri setindeki JPEG sıkıştırma ataklı görüntülere [35], [34] ve [38]' deki yöntemler ve önerilen

yöntem uygulanmıştır. Şekil 2.47 (a)' da 70 kalite faktörlü sıkıştırma atağına maruz kalmış sahte görüntülerden, Şekil 2.47 (b)'de ise 90 kalite faktörlü sıkıştırma atağına maruz kalmış görüntülerden elde edilen sonuçlarla çizilen ROC eğrileri gösterilmiştir. KF değeri 70 ve 90 olması durumunda bile önerilen yöntemden elde edilen sonuçlara göre çizilen ROC eğrisi iki durumda da ideal nokta olan 1'e daha yakıdır. Buradan yola çıkarak önerilen yöntemin JPEG sıkıştırma atağı durumunda diğer yöntemlere göre üstün performansa sahip olduğu söylenebilir.



(a)

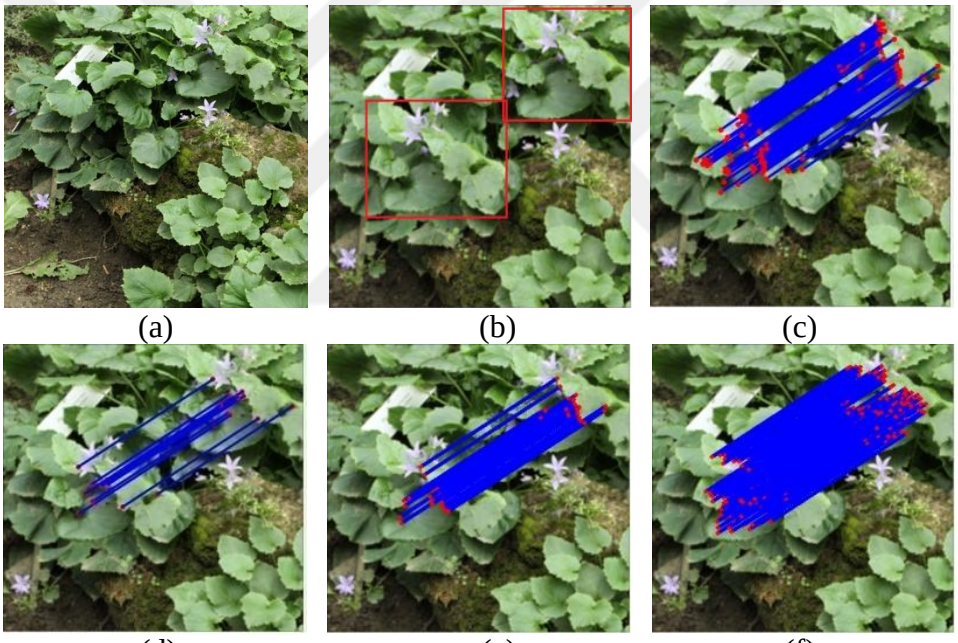


(b)

Şekil 2.47. (a) KF=70 ile JPEG sıkıştırma atağı durumunda (b) KF=90 ile JPEG sıkıştırma atağı durumunda SIFT [35], SURF [34], ORB [38] ve önerilen yöntemin ROC Eğrisi

2.6.6.3. Gauss Bulanıklaştırma Atağı Altındaki Deneysel Sonuçlar

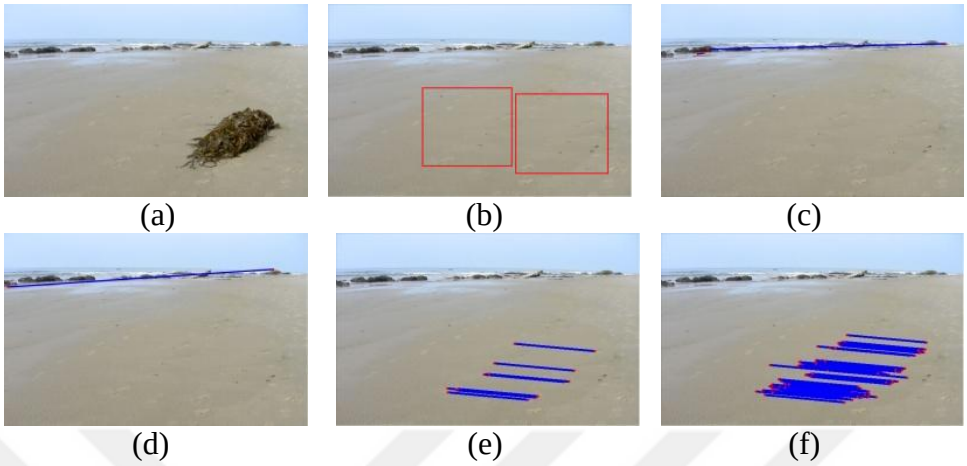
Kopyala yapıştır sahteciliği tespitinde bulanıklaştırma atağına maruz kalmış sahte görüntülerin tespit edilmesine ilişkin performans değerlendirmesi bu bölümde gerçekleştirilmiştir. Şekil 2.48 (a)'daki görüntüden bir bölgenin kopyalanıp yapıştırılarak daha sonra *3x3* pencere boyutu ve *σ=2* değerleriyle Gauss bulanıklaştırma işlemi uygulanarak Şekil 2.48 (c)' deki sahte görüntü elde edilmiştir. SIFT tabanlı [35]' deki yöntem ile *33*, SURF tabanlı [34]' deki yöntem ile *14* ve ORB tabanlı [38]' deki yöntem ile *19* adet eşleşme bulunmuştur. Elde edilen görsel sonuçlar sırasıyla Şekil 2.48 (c), (d) ve (e)' de görülmektedir. AKAZE tabanlı önerilen yöntemde ise *59* adet eşleşme bulunmuştur. Önerilen yöntemin eşleşme sonucu Şekil 2.48 (f)'de gösterilmiştir.



Şekil 2.48. (a) Orijinal görüntü(b) σ=2 ile Gauss bulanıklaştırma ataklı sahte görüntü (c) SIFT [35] sonucu (Doğru eşleşme sayısı:33) (d) SURF [34] sonucu (Doğru eşleşme sayısı:14) (e) ORB [38] sonucu (Doğru eşleşme sayısı:19) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:59)

Bulanıklaştırma atağı analizi için yapılan ikinci uygulamada ise nesne kapama işlemi gerçekleştirilen sahte görüntüye bir de *3x3* pencere boyutu ve *σ=0. 5* parametreleri ile Gauss bulanıklaştırma atağı uygulanarak Şekil 2.49 (b)'deki sahte görüntü elde edilmiştir. Bu görüntüye uygulanan SIFT [35] ve SURF [34] yöntemleri ile doğru eşleşme bulunamaz

iken ORB [38] yöntemi ile *9* adet eşleşme bulunmuştur. Önerilen yöntem ile de bu eşleşme sayısı *41*'e yükselmiştir.

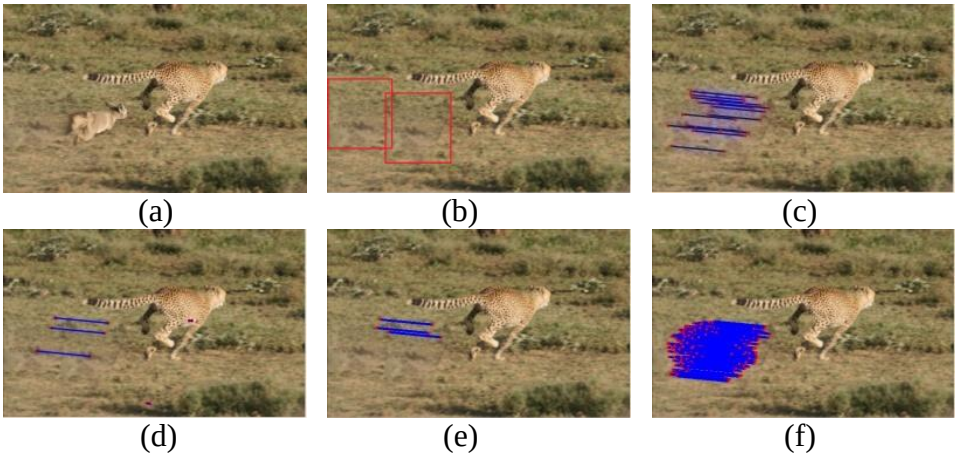


(a) (b) (c)

(d) (e) (f)

Şekil 2.49. (a) Orijinal görüntü (b) σ=0. 5 ile Gauss bulanıklaştırma ataklı sahte görüntü (c) SIFT [35] sonucu (Doğru eşleşme sayısı:0) (d) SURF [34] sonucu (Doğru eşleşme sayısı:0) (e) ORB [38] sonucu (Doğru eşleşme sayısı:9) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:41)

Şekil 2.50 (a)' daki görüntüye ise kopyala yapıştır işleminden sonra *3x3* pencere boyutu ve *σ=2* parametreleri ile Gauss bulanıklaştırma atağı uygulanarak Şekil 2.50 (b)' deki sahte görüntü oluşturulmuştur. Bu görüntüye uygulanan SIFT [35] ile 15, SURF [34] ve ORB [38]' den ise *3*'er adet eşleşme tespit edilmiştir. Önerilen yöntem ile de bu eşleşme sayısı *84*'e yükselmiştir.

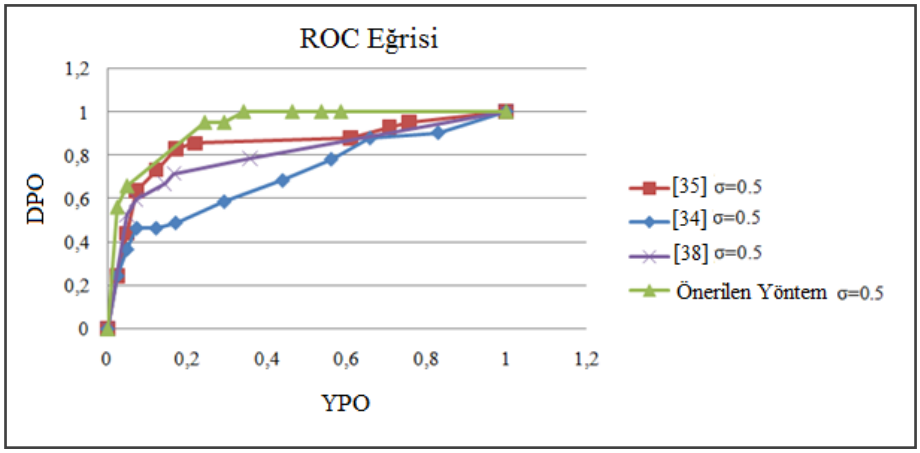


(a) (b) (c)

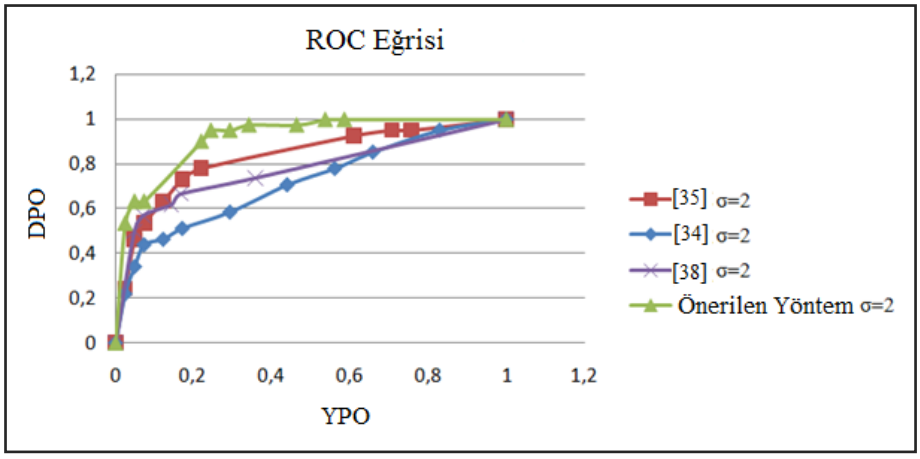
(d) (e) (f)

Şekil 2.50. (a) Orijinal görüntü (b) σ=2 ile Gauss bulanıklaştırma ataklı sahte görüntü (c) SIFT [35] sonucu (Doğru eşleşme sayısı:15) (d) SURF [34] sonucu (Doğru eşleşme sayısı:3) (e) ORB [38] sonucu (Doğru eşleşme sayısı:3) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:84)

Gauss bulanıklaştırma atağına karşı yöntemlerin gösterdiği performans sonuçlarının değerlendirilebilmesi için veri setindeki bulanıklaştırılmış sahte görüntülere [35], [34] ve [38]'deki yöntemler ve önerilen yöntem uygulanmıştır. Şekil 2.51 (a)'da *3x3* pencere boyutu ve *σ=0.5* parametreleri ile Gauss bulanıklaştırma atağına maruz kalmış sahte görüntülerden, (b)'de ise *3x3* pencere boyutu ve *σ=2* parametreleri ile Gauss bulanıklaştırma atağına maruz kalmış görüntülerden elde edilen sonuçlarla çizilen ROC eğrileri gösterilmiştir. *σ=0.5 ve σ=2* ile bulanıklaştırma atağına uğramış görüntülerin sahtecilik tespitinde bile önerilen yöntemden elde edilen sonuçlara göre çizilen ROC eğrileri ideal nokta olan 1'e daha yakındır. Bu eğrilerden yola çıkarak önerilen yöntemin diğer yöntemlere göre bu iki durumda da üstün performansa sahip olduğu söylenebilir.



(a)



(b)

Şekil 2. 51. (a) $\sigma = 0.5$ iken Gauss bulanıklaştırma atağı durumunda (b) $\sigma = 2$ iken Gauss bulanıklaştırma atağı durumunda SIFT [35], SURF [34], ORB [38] ve önerilen yöntemin ROC Eğrisi

2.6.6.4. AWGN Atağı Altındaki Deneysel Sonuçlar

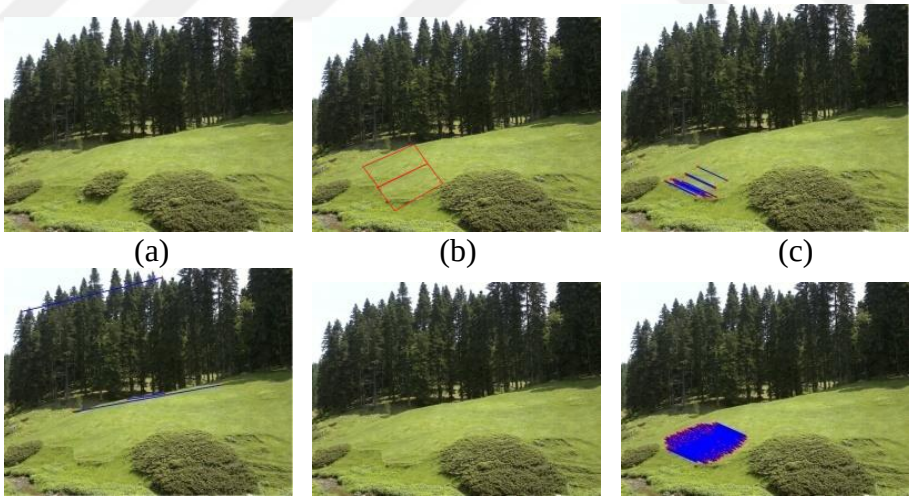
Yapılan son test işleminde kopyalanıp yapıştırılan görüntülere AWGN atağı uygulanması durumunda literatürdeki yöntemler ile önerilen yöntemin sahtecilik tespitine ilişkin performans analizi gerçekleştirilmiştir. Şekil 2.52 (a)' daki görüntüden bir bölgenin kopyalanıp yapıştırılarak daha sonra *25 dB* ile AWGN atağı uygulanarak Şekil (c)' deki sahte görüntü elde edilmiştir. SIFT tabanlı [35]'deki yöntem ile *79*, SURF tabanlı [34]'deki yöntem ile *29* ve ORB tabanlı [38]' deki yöntem ile *25* adet eşleşme bulunmuştur. Elde edilen görsel sonuçlar sırasıyla Şekil 2.52 (c), (d) ve (e)'de görülmektedir. AKAZE tabanlı önerilen yöntemde ise *143* adet eşleşme bulunmuştur. Önerilen yöntemin eşleşme sonucu ise Şekil Şekil 2.52 (f)'de gösterilmiştir.

(a) (b) (c)

(d) (e) (f)

Şekil 2.52. (a) Orijinal görüntü (b) 25 dB ile AWGN sıkıştırma ataklı sahte görüntü (c) SIFT [35] sonucu (Doğru eşleşme sayısı:79) (d) SURF [34] (Doğru eşleşme sayısı:29) sonucu (e) ORB [38] sonucu (Doğru eşleşme sayısı:25) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:143)

Şekil 2.53 (a)'daki görüntüye ise nesne kapama amaçlı kopyala yapıştır işleminden sonra 40 dB sinyal ile AWGN gürültü ekleme atağı uygulanarak Şekil 2.53 (b)'deki sahte görüntü oluşturulmuştur. Bu görüntüye uygulanan SIFT [35] ile *15* eşleşme bulunurken SURF [34] ve ORB [38] yöntemleri ile doğru eşleşme bulunamamıştır. Önerilen yöntem ile ise *119* adet eşleşme bulunmuştur. Bu etkin eşleştirme sonucu Şekil 2.53 (f)'de verilmiştir.



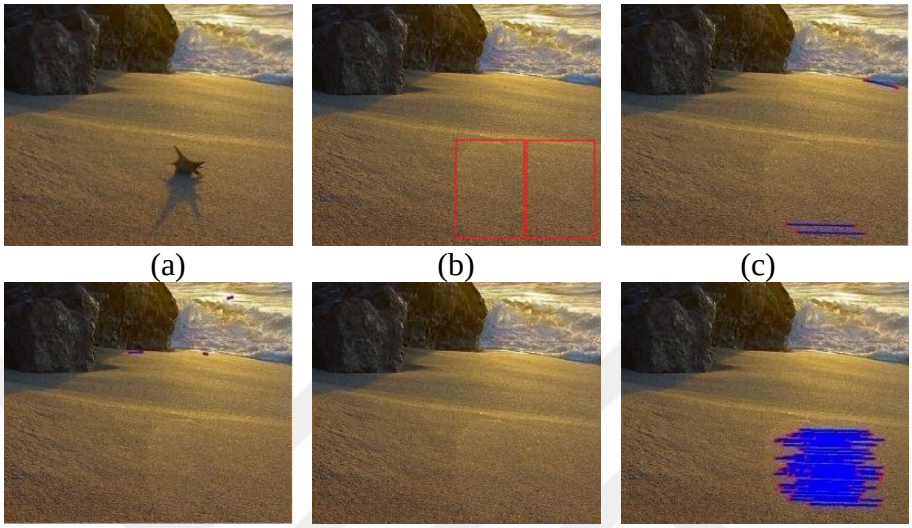
(a) (b) (c)

(d) (e) (f)

Şekil 2. 53. (a) Orijinal görüntü (b) 40 dB sinyal ile AWGN ataklı sahte görüntü (c) SIFT [35] sonucu (Doğru eşleşme sayısı:15)(d) SURF [34] (Doğru eşleşme sayısı:0) sonucu (e) ORB [38] sonucu (Doğru eşleşme sayısı:0) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı:119)

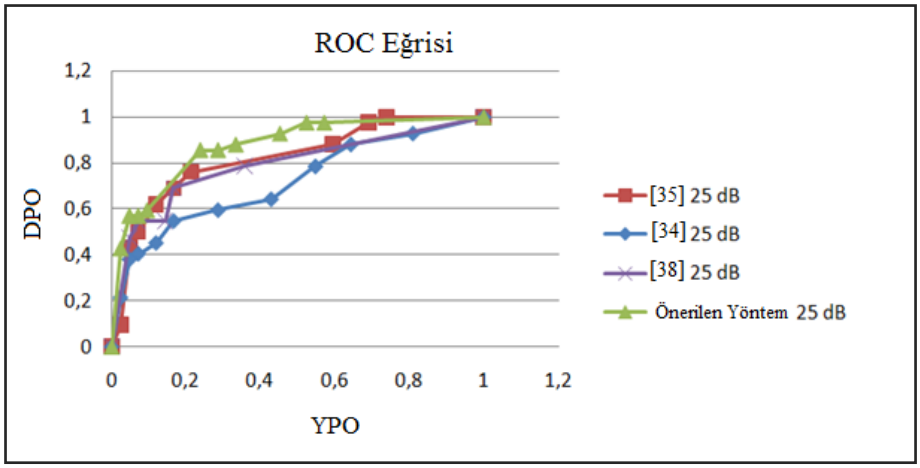
Son örnek görsel sonucu verilen test işleminde ise Şekil 2.54 (a)'daki görüntüye ise nesne kapama amaçlı kopyala yapıştır işleminden sonra *20 dB* sinyal ile AWGN gürültü

ekleme atağı uygulanarak Şekil 2.54 (b)'deki sahte görüntü oluşturulmuştur. Bu görüntüye uygulanan SIFT [35] ile *2* eşleşme bulunurken SURF [34] ve ORB [38] yöntemleri ile doğru eşleşme bulunamamıştır. Önerilen yöntem ile ise bu durumda bile *84* adet eşleşme bulunmuştur. Bu eşleştirme sonucu Şekil 2.54 (f)'de verilmiştir.

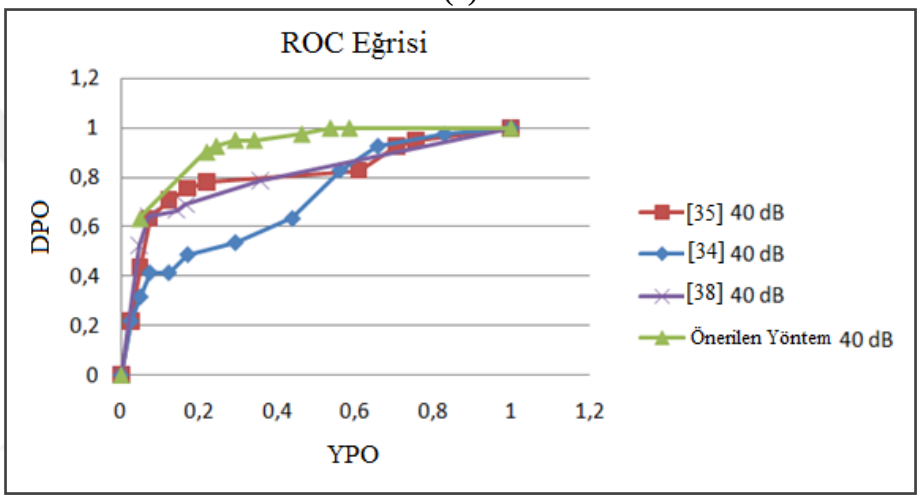


Şekil 2.54. (a) Orijinal görüntü (b) 20 dB sinyal ile AWGN ataklı sahte görüntü (c) SIFT [35] sonucu (Doğru eşleşme sayısı:2) (d) SURF [34] (Doğru eşleşme sayısı:0) sonucu (e) ORB [38] sonucu (Doğru eşleşme sayısı:0) (f) Önerilen yöntem sonucu (Doğru eşleşme sayısı :84)

Son analizin genel performans sonuçlarının değerlendirilebilmesi için veri setindeki bulanıklaştırılmış sahte görüntülere [35], [34] ve [38]'deki yöntemler ve önerilen yöntem uygulanmıştır. Şekil 2.55 (a)'da *25 dB* sinyali ile AWGN atağına maruz kalmış sahte görüntülerden, Şekil 2.55 (b)'de ise *40 dB* sinyali ile AWGN atağına maruz kalmış görüntülerden elde edilen sonuçlarla çizilen ROC eğrileri gösterilmiştir.



(a)



(b)

Şekil 2.55. (a) 25 dB sinyali ile AWGN atağı durumunda (b) 40 dB sinyali ile AWGN atağı durumunda SIFT [35], SURF [34], ORB [38] ve Önerilen yöntemin ROC Eğrisi

3. SONUÇLAR ve TARTIŞMA

Yapılan tez çalışmasında anahtar noktası tabanlı kopyala yapıştır sahteciliği tespitine ilişkin literatürde var olan yöntemlerin iyileştirilmesi yapılmış ve ayrıca yeni bir kopyala yapıştır sahteciliği tespiti önerilmiştir.

Tez kapsamında yapılan çalışmalara SURF tabanlı [34]'deki yöntemin RGB renk kanalları kullanılarak iyileştirilmesi gerçekleştirilmiştir. Bu yöntemin dönme, Gauss bulanıklaştırma ve AWGN ataklarına maruz kalması durumunda bile [34]'e göre üstün performansı deneysel sonuçlarda verilmiştir.

Daha sonra yapılan çalışmalarda literatürde var olan anahtar noktası tabanlı yöntemlerden [34,35] ve [38]'in uygulaması yapılarak SIFT ve SURF tabanlı yöntemlerde kullanılan Gauss ölçek uzayından dolayı ORB'de ise FAST algoritmasından dolayı özellikle düz bölgelerde yeteri kadar anahtar noktası tespit edilemediği gözlemlenmiştir [34, 35, 38]. Bu problemi çözebilmek için görüntüden öncelikle doku bilgisinin çıkarılması ön işlemi önerilmiştir. Yapılan ilk çalışmada LPQ ile doku bilgisi elde edilen görüntüden SIFT ile anahtar noktaları çıkarılarak sahtecilik tespiti gerçekleştirilmiştir. Yapılan bu çalışmanın JPEG sıkıştırma, Gauss bulanıklaştırma ve AWGN atakları durumunda bile [35]'e göre üstünlüğü hem görsel sonuçlar ile hem de TO metriği ile gösterilmiştir. Daha sonra ORB' de FAST algoritmasın daha etkin bir şekilde anahtar noktası tespit edebilmesi için Gabor filtresi kullanılarak doku görüntüsü elde edilmiş ve daha sonra sahtecilik tespiti gerçekleştirilmiştir. Bu iyileştirme ile elde edilen performans üstünlüğü görsel sonuçlarda ve TO metriği sonuçlarında gözlemlenmektedir.

Tez çalışmalarının devamında doku çıkarma ön işlemin sebep olacağı ek hesaplama karmaşıklığını ortadan kaldırmak için AKAZE tabanlı bir kopyala yapıştır sahteciliği tespiti yöntemi önerilmiştir. Bu yöntemin düz bölgelerde bile doku çıkarmak gibi bir ön işleme ihtiyaç duymadan tek başına kullanılabilecek kadar etkili olduğu sonuçlarla ortaya konulmuştur. Ayrıca dönme, Gauss bulanıklaştırma, JPEG sıkıştırma ve AWGN atakları durumunda bile [34,35] ve [38]'deki çalışmalara göre etkin sonuç ürettiği görsel sonuçlar ve ROC eğrileri ile ispatlanmıştır.

Tez kapsamında önerilen dört çalışmanın doku çıkarma ön işlemi içermesi, düz bölgelerle yapılan nesne kapama amaçlı gerçekleştirilen sahtecilikleri tespit edebilmesi ve yöntemin bulanıklaştırma, gürültü, JPEG sıkıştırma ve dönme ataklarına karşı dayanıklılık

durumlarını içeren bir değerlendirme Tablo 3. 1'de verilmiştir . Genel özellikleri göz önüne alındığında AKAZE tabanlı yöntem, ön işlem gerektirmeden nesne kapama sahteciliğini tespit edebilmesi ve ataklara karşı dayanıklılığı nedeniyle, diğer yöntemlere nazaran ön plana çıkmaktadır.

Tablo 3. 1. Önerilen yöntemlerin genel kıyaslaması

	1.Renkli SURF	2.LPQ ve SIFT	3. Gabor ve ORB	4. AKAZE
Doku Çıkarma Ön İşlemi	x	✓	✓	x
Düz Bölgelerde Sahtecilik Tespiti	x	✓	✓	✓
Gauss Bulanıklaştırma Atağına Dayanıklılık	✓	✓	✓	✓
AWGN Atağına Dayanıklılık	✓	✓	x	✓
JPEG Sıkıştırma Atağına Dayanıklılık	x	✓	✓	✓
Dönme Atağına Dayanıklılık	✓	x	x	✓

Yapılan LPQ ve SIFT tabanlı kopyala yapıştır sahteciliği tespiti yöntemine dayanan 1 adet bildiri Uluslararası Elektrik Elektronik ve Biyomedikal Mühendisliği Konferansında (ELECO' 2015) sözlü olarak sunulmuş ve bildiri kitapçığında basılmıştır [60].

Yapılan Gabor filtresi ve ORB tabanlı kopyala yapıştır sahteciliği tespiti yöntemine dayanan 1 adet bildiri International Conference on Image Processing, Production and Computer Science (ICIPCS' 2016) konferansında sözlü olarak sunulmuş ve bildiri kitapçığında basılmıştır [61].

Önerilen AKAZE tabanlı kopala yapıştır sahteciliği yönteminin sunulduğu bir yayın 7 Nisan 2016 tarihinde SCI Expanded kapsamındaki Mathematical Problems in Engineering Dergisine gönderilmiş olup under review durumundadır.

4. ÖNERİLER

AKAZE tabanlı yöntemin ayna yansıması afin dönüşümlere karşı da dayanıklı olduğu bir yöntem geliştirilebilir.

Kopyala yapıştır sahteciliği tespiti için blok tabanlı ve anahtar noktası tabanlı yöntemlerin birleştirilmesiyle görüntünün yapısal durumuna göre hibrit dinamik bir yöntem ile sahtecilik tespiti yapılabilir.

5. KAYNAKLAR

1. Qureshi, M. A. ve Deriche, M., A bibliography of Pixel-Based Blind Image Forgery Detection Techniques, Signal Processing: Image Communication, 39 (2015) 46–74.

2. Lian, S. ve Kanellopoulos D., Recent Advances in Multimedia Information System Security, Informatica, 33 (2009) 3–24.

3. Rey, C. ve Dugelay, J. L., A Survey of Watermarking Algorithms For Image Authentication, EURASIP Journal on Applied Signal Processing, (2002) 613–621.

4. Kundur, D. ve Hatzinakos, D., Digital Watermarking for Telltale Tamper Proofing and Authentication, Proceedings of the IEEE, 87, 7 (1999) 1167–1180.

5. Om, A. ve Be, K., Passive detection of Copy-Move Forgery in Digital Images: State-of-the-Art, Forensic Science International, 231 (2013) 284–295.

6. Zhang, Z., Zhou, Y., Kang, J., ve Ren, Y., Study of Image Splicing Detection, Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues, 5226 (2008) 1103-1110.

7. http://forensics.idealtest.org, 6 Ekim 2015.

8. Redi, J. A., Taktak, W. ve Dugelay, J. L., Digital Image Forensics: A Booklet for Beginners, Multimedia Tools Appl., 51, 1 (2011) 133–162.

9. Fridrich, A. J., Soukal, B. D. ve Lukáš, A. J., Detection of Copy-Move Forgery in Digital Images, Digital Forensic Research Workshop (DFRWS), 2003.

10. http://www.scopus.com, 5 Ağustos 2015.

11. Popescu, A. ve Farid, H., Exposing Digital Forgeries by Detecting Duplicated Image Regions, Tech. Rep., TR2004-515, Dartmount Collage, 2004.

12. Luo, W., Huang, J. ve Qiu, G., Robust Detection of Region-Duplication Forgery in Digital Images, International Conference on Pattern Recognition, Kasım 2009, Hong Kong, Bildiriler Kitabı 4: 746–749.

13. Mahdian, B. ve Saic, S., Detection of Copy-Move Forgery Using a Method Based on Blur Moment Invariants, Forensic Sci. Int., 171 (2007) 180–189.

14. Myrna, A. N., Venkateshmurthy, M. G. ve Patil, C. G., Detection of Region Duplication Forgery in Digital Images Using Wavelets and Logpolar Mapping, IEEE Int. Conf. Computational Intelligence and Multimedia Applications, Aralık 2007, Sivakasi, Tamil Nadu, Bildiriler Kitabı: 371–377.

15. Kang, X. ve Wei, S., Identifying Tampered Regions Using Singular Value Decomposition in Digital Image Forensics, International Conference on Computer Science and Software Engineering, Aralık 2008, Wuhan, Hubei, Bildiriler Kitabı: 926–930.

16. Zhang, J., Feng, Z, ve Su,Y., A New Approach for Detecting Copy-Move Forgeries in Digital Images, Intl. Conference on Communication Systems, Kasım 2008, Guangzhou, Bildiriler Kitabı: 362–366.

17. Bayram, S., Sencar, H. T. ve Memon, N., An Efficient and Robust Method For Detecting Copy-Move Forgery, IEEE International Conference on Acoustics, Speech and Signal Processing, Nisan 2009, New York, Bildiriler Kitabı: 1053 – 1056.

18. Wang, J., Liu, G., Li, H., Dai, Y. ve Wang, Z., Detection of Image Region Duplication Forgery Using Model With Circle Block, Intl. Conference on Multimedia Information Networking and Security, Kasım 2009, Hubei, Bildiriler Kitabı: 25–29.

19. Bashar, M., Noda, K., Ohnishi, N. ve Mori, K., Exploring Duplicated Regions in Natural Images, IEEE Transaction on Image Processesing, 99 (2010) 1–40.

20. Khan, S. ve Kulkarni, A., Reduced Time Complexity For Detection of Copy-Move Forgery Using Discrete Wavelet Transform, International Conference & Workshop on Emerging Trends in Technology, Şubat 2010, New York, Bildiriler Kitabı: 31–36.

21. Huang, Y., Lu, W., Sun, W. ve Long, D., Improved DCT Based Detection of Copy-Move Forgery in Images, Forensic Science International, 206 (2011) 178–184.

22. Bravo-Solorio, S. ve Nandi, A.K., Exposing Duplicated Regions Affected by Reflection, Rotation and Scaling, Intl. Conference on Acoustics, Speech and Signal Processing, Mayıs 2011, Prague, Bildiriler Kitabı: 1880–1883.

23. Gharibi, F., RavanJamjah, J., Akhlaghian, F ve Azami, B. Z., Robust Detection of Copy-Move Forgery Using Texture Features, 19th Iranian Conference on Electrical Engineering, Mayıs 2011, Tahran, İran, Bildiriler Kitabı: 1–4.

24. Hsu, H. ve Wang, M., Detection of Copy-Move Forgery Image Using Gabor Descriptor, Anti-counterfeiting, Security and Identification, Ağustos 2012, Taipei, Taiwan, Bildiriler Kitabı: 1–4.

25. Muhammad, G., Hussain, M. ve Bebis, G., Passive Copy Move Image Forgery Detection Using Undecimated Dyadic Wavelet Transform, Digital Investigation, 9, 1 (2012) 49–57.

26. Li, L., Li, S. ve Zhu, H., An Efficient Scheme for Detecting Copy-Move Forged Images by Local Binary Patterns, Journal of Information Hiding and Multimedia Signal Processing, 4, 1 (2013) 46–56.

27. Ryu, S., Kirchner, M, Lee, M. ve Lee, H., Rotation Invariant Localization of Duplicated Image Regions Based on Zernike Moments, IEEE Transaction on Information Forensics and Security , 8, 8 (2013) 1355–1370.

28. Lee, J., Chang, C. ve Chen, W., Detection of Copy–Move Image Forgery Using Histogram of Oriented Gradients, Information Sciences, 321 (2015) 250–262.

29. Lee, J., Copy-Move Image Forgery Detection Based on Gabor Magnitude, Journal of Visual Communication and Image Representation, 31 (2015) 320–334

30. Bi, X., Pun, C. ve Yuan, X., Multi-Level Dense Descriptor and Hierarchical Feature Matching for Copy–Move Forgery Detection, Information Sciences, 345 (2016) 226–242.

31. Huang, H., Guo, W. ve Zhang, Y., Detection of Copy-Move Forgery in Digital Images using SIFT Algorithm, Computational Intelligence and Industrial Application, Computer Society, Aralık 2008, Wuhan, Bildiriler Kitabı: 272–276.

32. Pan, X. ve Lyu, S., Detecting Image Region Duplication Using SIFT Features, International Conference on Acoustics, Speech and Signal Processing, Mart 2010, Dallas, Bildiriler Kitabı: 1706 – 1709.

33. Pan, X. ve Lyu, S., Region duplication detection using image feature matching, IEEE Transactions on Information Forensics and Security, 5 (2010) 857-867.

34. Xu, B., Wang, J., Liu, G., Li, H. ve Dai, Y., Image Copy-Move Forgery Detection Based on SURF, International Conference on Multimedia Information Networking and Security, Kasım 2010, Nanjing, Jiangsu, Bildiriler Kitabı: 889–892.

35. Amerini, I., Ballan, L., Caldelli, R., Bimbo, A.D. ve Serra, G., A SIFT-Based Forensic Method For Copy-Move Attack Detection and Transformation Recovery, IEEE Transactions on Information Forensics and Security, 6, 3 (2011) 1099–1110.

36. Jaberi, M., Bebis, G., Hussain, M. ve Muhammad, G., Improving the Detection and Localization of Duplicated Regions in Copy-Move Image Forgery, 18th International Conference on Digital Signal Processing (DSP), Temmuz 2013, Fira, Bildiriler Kitabı: 1-6.

37. Kiruthika, K., Mahalakshmi, S.D. ve Vijayalakshmi, K, Detecting Multiple Copies of Copy-Move Forgery Based on SURF, International Conference on Innovations in Engineering and Technology, Mart 2014, Tamil Nadu, Bildiriler Kitabı: 2347-6710.

38. Zhu, Y., Shen, X. ve Chen, H., Copy-Move Forgery Detection Based on Scaled ORB, Multimedia Tools and Applications, 75, 6 (2015) 1-15.

39. Wenchang, S., Fei, Z., Bo, Q. ve Bin, L., Improving Image Copy-Move Forgery Detection With Particle Swarm Optimization Techniques, China Communications, 13, 1 (2016) 139 - 149.

40. Lowe, D. G., Object Recognition From Local Scale-Invariant Features, International Conference on Computer Vision, Eylül 1999, Kerkyra, Bildiriler Kitabı: 1150-1157.

41. Bay, H., Ess, A., Tuytelaars, T. ve Van Gool, L., SURF: Speeded Up Robust Features, Computer Vision and Image Understanding, 110, 3 (2008) 346-359.

42. Rublee, E., Rabaud, V., ve Konolige, K., ORB: An Efficient Alternative to SIFT or SURF, International Conference on Computer Vision, Kasım 2011, Barcelona, Bildiriler Kitabı: 2564 – 2571.

43. Rosten, E. ve Drummond, T., Machine Learning For High Speed Corner Detection, Lecture Notes in Computer Science, 1 (2006) 430–443.

44. Rosten, E., Porter, R. ve Drummond, T., Faster and Better: A Machine Learning Approach to Corner Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32, 1 (2010) 105-119.

45. Calonder, M., Lepetit, V. ve Strecha, C., Brief: Binary Robust Independent Elementary Features, Lecture Notes in Computer Science, 6314 (2010) 778-792.

46. Ojansivu, V. ve Heikkilä, J., Blur Insensitive Texture Classification Using Local Phase Quantization, Lecture Notes in Computer Science, 5099 (2008) 236-243.

47. Daugman, J. G., Two-Dimensional Spectral Analysis of Cortical Receptive Field Profile, Vision Research, 20, 10 (1980) 847-856.

48. Fischler, M. A. ve Bolles, R. C., Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM, 24, 6 (1981) 381–395.

49. Alcantarilla, P. F., Bartoli, A ve Davison, A. J., KAZE Features, Lecture Notes in Computer Science, 7577 (2012) 214-227.

50. Alcantarilla, P. F., Nuevo, J. ve Bartoli, A., Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, In British Machine Vision Conference (BMVC), Eylül 2013, Bristol.

51. Perona, P. ve Malik, J., Scale-Space and Edge Detection Using Annisotropic Diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12, 7 (1990) 1651–1686.

52. Yang, X. ve Cheng, K.T., LDB: An Ultra-Fast Feature for Scalable Augmented Reality, IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Kasım 2012, Atlanta GA, Bildiriler Kitabı: 49-57.

53. Gonzalez, R. C. ve Woods, R.E, Digital Image Processing, Third Edition, Pearson Education International, 2008.

54. Metz, C. E, Receiver Operating Characteristic Analysis: A Tool for the Quantitative Evaluation of Observer Performance and Imaging Systems, Journal of the American Collage of Radiology, 3, 6 2006, 413-422.

55. Flach, P., ROC Analysis, Encyclopedia of Machine Learning (Springer), 80 (2010) 869-875.

56. Krzanowski, W. J. ve Hand, D. J., ROC Curves for Continuous Data, Boca Raton: Chapman and Hall/CRC Press., (2009).

57. http://images.google.com, 2 Aralık 2015.

58. http://www.vcl.fer.hr/comofod, 15 Ocak 2016.

59. Metz, E. C., Receiver Operating Characteristic Analysis: A Tool For The Quantitative Evaluation Of Observer Performance and Imaging Systems, Journal of the American Collage of Radiology, 3, 6 (2006) 413-422.

60. Ustübioglu, B., Muzaffer, G., Ulutas, G., Nabiyev, V., Ulutas, M., A Novel Keypoint Based Forgery Detection Method Based On LPQ and SIFT, International Conference on Electrical and Electronics Engineering, Ekim 2015, Bursa, Bildiriler Kitabı: 185 – 189.

61. Muzaffer, G., Makul, O., Ustübioglu, B., Ulutas, G., Copy Move Forgery Detection Using Gabor Filter and ORB, International Conference on Image Processing, Production and Computer Science, Mart 2016, Londra, Bildiriler Kitabı: 23-30.

ÖZGEÇMİŞ

Gül Muzaffer,1989 Kayseri doğumludur. İlkokulu Derviş Güneş İlkokulu ve ortaokulu Şehit Binbaşı Mahmut Şahin İlköğretim Okulu'nda ardından liseyi Fatma Kemal Timuçin Anadolu Lisesi'nde tamamlamıştır. Erciyes Üniversitesi Bilgisayar Mühendisliği Bölümü'nden 2012 yılında mezun olmuştur. Aynı yıl Fırat Üniversitesinde yüksek lisans eğitimine ve araştırma görevliliğine başlamıştır. Bu görevinden 2015 yılında Karadeniz Teknik Üniversitesine geçiş yapıp yüksek lisans eğitimine Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda devam etmiştir. Halen Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaktadır. İyi derecede İngilizce bilmektedir.