İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

UZAKTAN ÇEKİLMİŞ FOTOĞRAFLARLA MOZAİK OLUŞTURMA

YÜKSEK LİSANS TEZİ
Müh. Mesut PAK

Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Programı : BİLGİSAYAR MÜHENDİSLİĞİ

ŞUBAT 2008

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

UZAKTAN ÇEKİLMİŞ FOTOĞRAFLARLA MOZAİK OLUŞTURMA

YÜKSEK LİSANS TEZİ

Müh. Mesut PAK
(504051521)

Tezin Enstitüye Verildiği Tarih : 24 Aralık 2007
Tezin Savunulduğu Tarih : 29 Ocak 2008

Tez Danışmanı :	**Yrd.Doç.Dr. Turgay ALTILAR**
Diğer Jüri Üyeleri	**Prof.Dr. Muhittin Gökmen**
	Doç.Dr. Elif Karslıgil (Y.T.Ü.)

ŞUBAT 2008

ÖNSÖZ

Bu çalışmayı bana öneren ve iki yıl süren bu tez çalışmam boyunca teknik anlamda yardımlarını esirgemeyen hocam Sayın Yrd. Doç. Dr. Turgay ALTILAR'a, bu çalışmayı yapmak için bana fırsat tanıyan TÜBİTAK'a, test materyalleri sağladığı için BAYKAR MAKİNA şirketine, ömrümün ilk gününden beri her konuda destek olan anneme, babama, ruhumun diğer yarısı olan eşime, onun ailesine ve benden desteğini esirgemeyen herkese çok teşekkür ederim.

Ocak 2008 Mesut Pak

İÇİNDEKİLER

KISALTMALAR

UAV	: Unmanned Aerial Vehicle
İHA	: İnsansız Hava Aracı
SIFT	: Ölçekten Bağımsız Özellik Dönüşümü
Çİ	: Çapraz İlinti

TABLO LİSTESİ

ŞEKİL LİSTESİ

UZAKTAN ÇEKİLEN FOTOĞRAFLARLA MOZAİK OLUŞTURMA

ÖZET

Bu çalışmada bir Hava Aracı ile elde edilmiş resimlerin iki boyutlu düzlemdeki yerlerinin bulunmasıyla, bunların daha büyük bir resim içerisinde toplanması, başka bir deyişle bölgenin haritasının (mozaiğinin) çıkarılması amaçlanmıştır. Kullanıcı etkileşimini en aza indirerek otomatik olarak resimler arasındaki kayıklıkları hesaplayan ve tek bir resim içerisinde birleştiren bir uygulama kullanıcı arayüzü ile birlikte geliştirilmiştir. Bu uygulama, bu çalışmada önerilen resim kaynaştırma algoritmasının denenmesi amacıyla Visual Studio .NET C++ yazılım geliştirme ortamında yazılmıştır. Bu algoritmaya ulaşılmadan önce MATLAB ve Visual Studio .NET C++ ortamlarında iki uygulama daha geliştirilmiş, fakat o uygulamalardaki algoritmalar kullanışsız olduğu için arşive aktarılarak yeni çözümler aranmaya devam edilmiş ve bu tez çalışmasında sunulan algoritma ortaya çıkmıştır.

Yakından çekilen resimlerde, bilindiği gibi farklı noktalardan veya küçük açı değişimleriyle resimler çekildiğinde şekiller perspektif veya bakış açısı gibi sebeplerle farklılaşır. Bu durum uçaktan çekilen resimlerde nerdeyse hiç gözlenmemektedir. Çünkü uçakta bakış açısı veya perspektifin değişmediği kabul edilebilir bir şekilde resimler çekilmektedir. Başlangıçta genel amaçlı bir resim kaynaştırma algoritmasına yönelik çalışıldığı için yakından çekilen resimlerdeki perspektif farklılığı nedeniyle oluşan şekil farklılaşmalarını yok eden (görmezden gelen) bir yöntem geliştirilmiştir. Daha sonraları bu farklılaşmaların nerdeyse hiç gözlenmediği uçak resimlerinin kaynaştırılması olarak konunun özelleşmesine rağmen gene aynı yöntem kullanılmıştır.

Bu yöndeki geçmiş çalışmalara bakıldığında, herhangi iki resimdeki örtüşen kısımların tespiti, kaynaştırma işleminin temel parçasıdır. Dolayısıyla kaynaştırılmak üzere verilen resimlerde yeterli oranda örtüşen alanların olduğu varsayılmaktadır. Örtüşme kavramı, iki farklı resim alanının eşleştirilmesini de içerdiğinden şekil

tanıma alanına da dallanılmıştır. Şekil tanıma kısmında alana dayalı yönteme de benzeyen ancak özellik tabanlı bir yöntem olan (shape context) şekil içerik yapısı yöntemi kullanılmıştır. Bu şekil içerik yapılarının karşılaştırılması, çapraz ilinti (cross correlation) yöntemiyle yapılmıştır. Ayrıca karşılaştırmanın hızını artırmak için kenar yoğunluk matrisi kullanılmıştır. Var olan çalışmaların büyük bir bölümünde kullanıcının benzerlikleri vermesi veya oluşturulmuş mozaik resmin düzenlenmesi gibi işlemleri yapması istenirken otomatik olarak resimleri kaynaştıran yöntemler oldukça azdır. Resim adresleme, alan mozaik çıkarma, resim dikiş yapma, resim ulama gibi isimlerle benzer çalışmalar yapılmış olan bu çalışmada otomatik bir çözüm geliştirilmiştir.

IMAGE MOSAICING BY USING PHOTOGRAPHS TAKEN FAR AWAY

SUMMARY

In this work the aim is obtaining the map (mosaic) of an area by registering images acquired from an Aerial Vehicle in a large image. To register images in a large image, their locations in the large image must be calculated. A computer program containing a Graphical User Interface that calculates shiftings between images and registers them automatically in a large image, also decreases user interaction, is developed. This application is written in Visual Studio .NET C++ development environment to experience the image registering algorithm that proposed in this work. Before this algorithm be reached, two more algorithms were developed and experienced in MATLAB and VS .NET C++ environments, however because of their badly performance and very low speed they were put into archive and the study went on with searching for new solutions until reaching the work presented in this thesis study.

In images those taken from a small distance, as it is known that shapes become different due to view angle or perspective differences when images are taken from different view points or with small angle differences. This statement is not valid for images taken from UAVs in general because of the camera-area distance. Being this distance long, in UAVs the images are acquired with assumption that there is no change in viewing angle and perspective. At the beginning, because this study was going around a general purpose image stitching algorithm, a method that ignores (gets rid of) the shape changes in perspective or view angle is developed. Although the study become special as registering image acquired with aerial vehicles in which this perspective or view angle changes are not visible, the same method is still used.

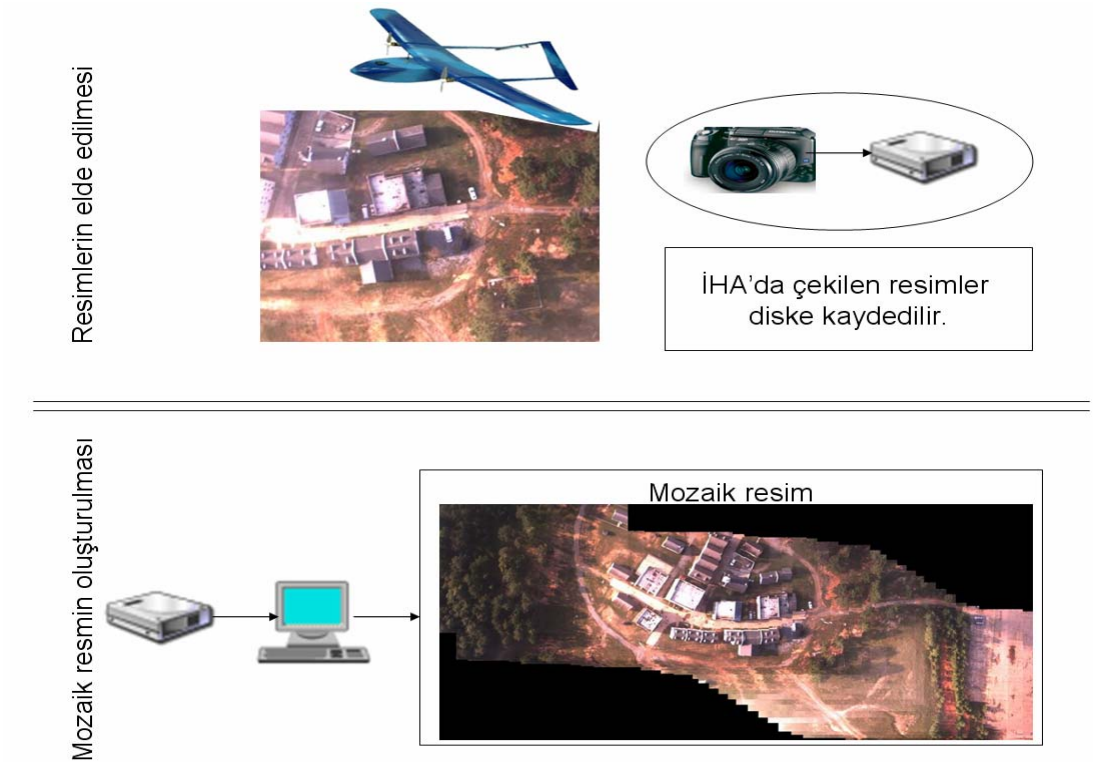
When looked at recent studies, deciding the overlapped area of two images is the main part of the mosaic operation. For this reason, it is assumed that there is enough overlapped area in any couple of images. To decide overlapped area, several parts of

two different images must be matched; therefore, shape recognition work is issued in this study. In shape recognition part, a method that looks like an area based method but it is a feature based method, shape context definition, is used. Cross correlation algorithm is used for comparing the shape context structures. Additionally, edge density matrix is prepared in order not to do meaningless comparison, so the speed of mosaic is increased considerably. While present mosaic studies need user interaction such as marking control points between images and editing the mosaic image, automated mosaic studies are too few. In mosaicing area; also named as image registering, mosaicing, image stitching and making panorama; an automated solution is designed and developed.

1. MOZAİK VE İLGİLİ KAVRAMLAR

1.1. Mozaik İşleminin Tanımı

Bir bölgede belirli bir yükseklikten fotoğraf çekilerek elde edilen resimlerin tek bir mozaik resim içerisinde adreslenmesiyle o bölgenin haritasının oluşturulmasına mozaik çıkarma denir. Mozaik işleminin aşamaları **Şekil 1.1**'de görülmektedir. Mozaik çıkarma, resim adresleme, resimlerin sicile geçirilmesi, resimleri dikiş yapma gibi isimlerle üzerinde pek çok yöntem geliştirilmiş bir konudur.



Şekil 1.1: Mozaik İşleminin Aşamaları

1.2. Ayrıt Saptama

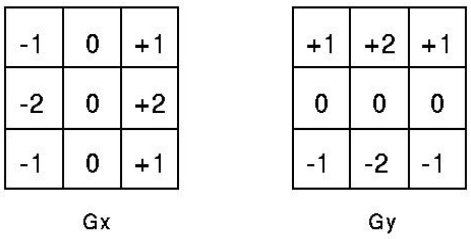
İki fotoğraf içerisindeki karşılıklı iki pencere arasındaki benzerliğin hesaplanması için Şekil İçerik yapısı kullanılmaktadır. Bu yapı fotoğrafın ayrıt haritası üzerinde tanımlıdır. Bundan dolayı fotoğraflar üzerinde yapılan ilk işlem ayrıt saptamadır.

Ayrıt, resimdeki cisimlerin kenarları, çizgisel şekiller, ani renk geçişleri gibi yerleri belirtir. Bir resmin ayrıtlarının saptanması, resimdeki önemli yapısal özellikleri ortaya çıkarırken, resimdeki bilgiyi önemli ölçüde azaltır ve gereksiz bilgileri filtreler [2]. Renkli olarak çekilen fotoğraflar siyah-beyaz resimlere dönüştürüldükten sonra gri tonlarda ani geçişlerin olduğu bölgeler bulunur. Elde edilen resim, en basit haliyle bir resmin ayrıt haritasıdır. Ayrıt haritası çıkarma işleminin ayrıt olmayan yerleri ayrıt olarak işaretlememesi ve ayrıt olan yerleri de ayrıt olarak işaretlemesi gerekmektedir. Bunun yanında saptayıcı ayrıt olan yerlerin konumlarını da doğru işaretlemelidir. Başka bir deyişle bir ayrıt, bir ayrıt olarak işaretlenmeli ve ayrıtın kalınlığı en az olmalıdır.

$$\frac{1}{115}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Şekil 1.2: Gaussian filtre maskesi görülmektedir ($\sigma = 1.4$).

Bu çalışmada Canny Ayrıt Saptayıcısı kullanılmıştır. Canny Ayrıt Saptayıcısı gri tonlar üzerinde tanımlıdır. Gri tonlara dönüştürülmüş olan resim bu saptayıcıya girer ve giriş resmiyle aynı boyutlarda ikili (binary) bir resim çıkar. Resimdeki gürültüyü kaldırmak için Gauss Filtresinden geçirilen resimdeki uzaysal türevi yüksek olan yerlerin işaretlenmesi için resimde her piksel için eğim (gradyan) hesaplanır. Gauss filtresindeki maskenin genişliği arttıkça ayrıt saptayıcısının konum hatası artar. Gaussian filtresi için kullanılan filtre maskelerinden biri Şekil 1.2'de verilmiştir. Bu Gaussian filtresi varyansı 1.4 alınarak hesaplanmıştır. Farklı Gaussian maskeleri MATLAB ortamında "fspecial" fonksiyonu kullanılarak elde edilmiş ve kullanılmıştır. Bu maskelerin varyans değerleri 0.5, 0.75, 0.9, 1.0, 1.25, 1.4, 1.75 ve 2.0 olarak seçilmiştir. Gri ton resimdeki ayrıt eğiminin hesaplanması için yatay ve düşey eksenlerde kullanılan iki ayrı maske Şekil 1.3'te görülmektedir. Hem Gauss filtresi hem de eğim hesabı, resim ve maskenin birbiriyle konvolüsyonu ile gerçekleştirilir.



Şekil 1.3: Ayrıt eğimi hesaplamada kullanılan maskeler görülmektedir.

Eğim hesabında her bir piksele karşılık bir ayrıt şiddeti ve ayrıt doğrultusu bulunur. Her bir piksele ait ayrıt şiddeti Denklem 1.1 ile hesaplanır.

$$\|G\| = |G_x| + |G_y| \tag{1.1}$$

Denklem 1'deki G_x bir noktadaki yatay eğimi ve G_y bir noktadaki düşey eğimi belirtir. $|G_x|$ ve $|G_y|$ bunların mutlak değeridir ve $\|G\|$ ise toplam ayrıt şiddetidir. Bir ayrıtın doğrultusu Denklem 1.2 ile hesaplanır. Ayrıca ayrıt doğrultusu hesaplanırken sıfıra bölme durumu göz önünde bulundurulur.

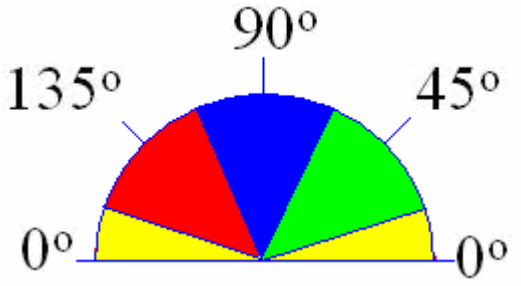
$$\alpha = \text{invtan}\left(G_x / G_y\right) \tag{1.2}$$

Bir noktada bir ayrıtın doğrultusu 4 farklı değerden birini alabilir. Örneğin Şekil 1.4'te olduğu gibi 5x5'lik bir resim parçasında 'a' harfi bir pikseli, etrafındaki 'x' harfleri ise komşularını göstersin.

$$\begin{vmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & a & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{vmatrix}$$

Şekil 1.4: Bir ayrıt pikseli ve komşuları görülmektedir.

Burada bahsedilen 'a' ayrıtı, 0° (yatay doğrultu), 45° (pozitif köşegen), 90° (düşey doğrultu) veya 135° (negatif köşegen) doğrultularından birine sahip olabilir. Ara değerler olan 20°, 85° gibi doğrultularda bir ayrıt olamaz. Bundan dolayı hesaplanan doğrultu değeri bu 4 doğrultudan biriyle eşleştirilir.



Şekil 1.5: Ayrıt piksellerinin ayrıt doğrultuları 4 alanda incelenir.

Bundan dolayı, sarı alandaki (0° – 22.5° & 157.5° – 180°) açılar 0°, yeşil alandaki (22.5° – 67.5°) açılar 45°, mavi alandaki (67.5° – 112.5°) açılar 90°, son olarak kırmızı alandaki (112.5° – 157.5°) açılar 135° doğrultusunda olarak kabul edilir.

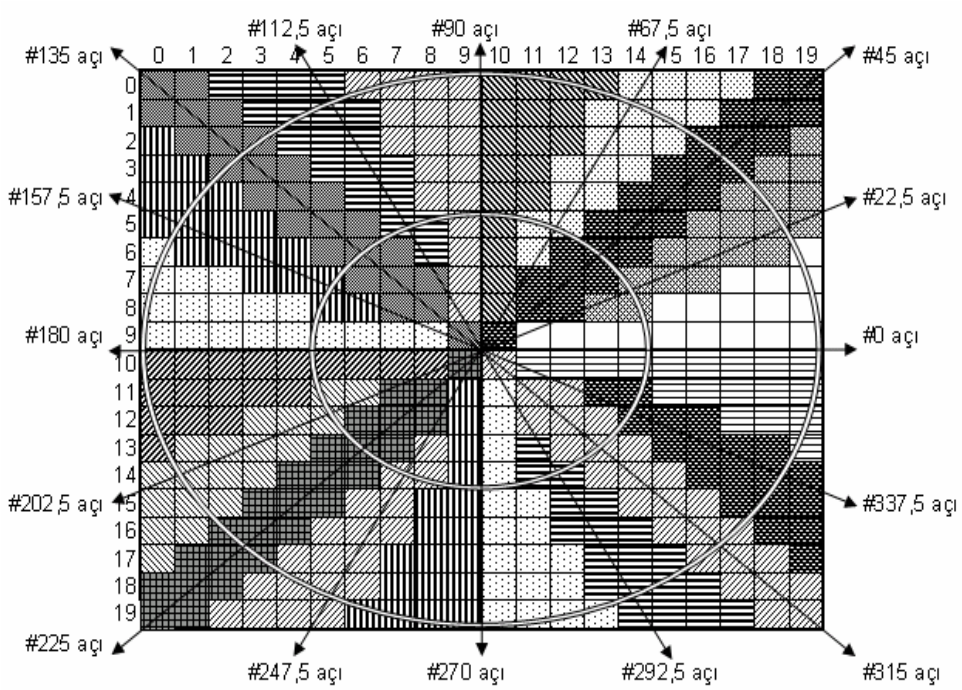
Bir sonraki aşamada her bir pikseldeki ayrıt açısı ve şiddeti kullanılarak ayrıt olarak işaretlenemeyecek olan pikseller tespit edilir.

Son aşamada iki eşik değeri kullanılarak ayrıtlara ait pikseller belirlenir. Eğim değeri büyük eşikten yukarı olan pikseller ayrıt piksel olarak atanırken, küçük eşik ile büyük eşik arasında eğime sahip pikseller eğer bir ayrıt pikseline komşu ise ayrıt piksel olarak atanır. İki eşikli bir karar mekanizmasının nedeni, gürültü faktöründen dolayı ayrıt üzerindeki bazı noktaların eğim değerinin düşük kalmasıdır.

1.3. Şekil Tanımlama

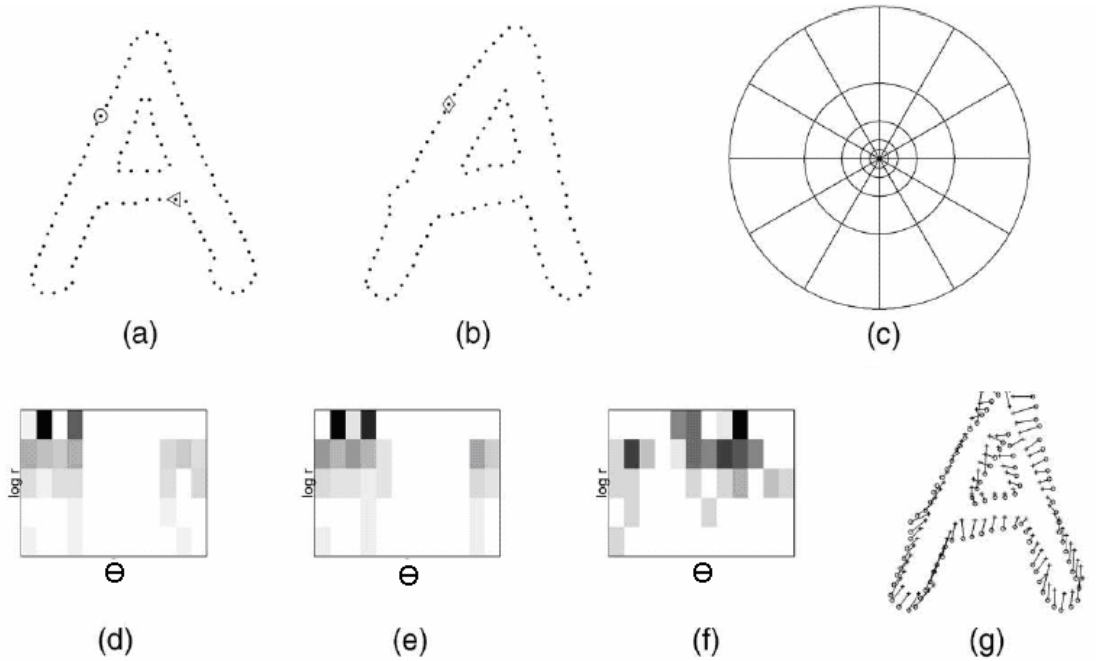
Şekillerin tanımlanmasında özellik tabanlı ve alan tabanlı olmak üzere iki yöntem uygulana gelmiştir. Resimdeki ayrıtlar, köşe noktaları, resimde bulunan nesneler gibi çeşitli özelliklerin saptanarak bunların eşleştirilmesi özellik tabanlı yöntemlerde yapılır. Alan tabanlı yöntemlerde ise saf yoğunluk denilen gri ton değerleri kullanılır.

Özellik tabanlı bir yöntem olan şekil içeriği yapısı (shape context), üzerinde çalışılan son şekil tanımlama fikridir. İnsan gözünün şekilleri algılamasından esinlenerek oluşturulan bu yöntemde şeklin merkezine odaklanılarak etrafa belirli açılarla ışınlar çizilir ve merkezden itibaren belirli çaplarda halkalar çizilir. Sonuçta şekil kutulara bölünmüş olur. Bu çalışmada 20 dilim ve 2 halka kullanılarak her bir pencere 40 kutuya bölünmüştür.



Şekil 1.6: 2 halka ve 20 dilim kullanılarak oluşan bir Şekil İçerik Yapısı görülmektedir.

Şekil içerik yapısı kullanılarak 40 kutuya parçalanan pencerede, her bir kutuya düşen ayrıt piksel sayısı o pencerenin şekil içerik bağlamında tanımıdır. Bu şekil içerik yapıları karşılaştırılarak pencerelerin benzerlikleri hesaplanır. Karşılaştırma işleminde çapraz ilinti (cross correlation) yöntemi kullanılır.



Şekil 1.7: Şekil İçeriği hesaplanması ve yaklaşıklık mantığı görülmektedir.

Şekil içeriği hesaplanması ve bu hesaplamadaki yaklaşıklık mantığı **Şekil 1.7**'de görülmektedir. Şekil içeriği hesaplanmasında bir merkez noktası seçilerek (a ve b şekillerindeki ○,◊ ve Δ noktaları) elde edilen yapı (c şeklindeki yapı) için her doğrultu ve halkadaki ayrıt piksel yoğunlukları (d, e ve f) görülmektedir. **Şekil 1.7**'deki (c) kısmında görüldüğü gibi 12 doğrultu ve 5 halkalı bir yapı örneği verilmiştir. Bu çalışmada ise 20 doğrultu ve 2 halka kullanılarak şekil içerik yapısı 40 kutuya bölünmüştür.

Şekil içerik yapısı ile şekillerin karşılaştırılmasındaki en önemli avantaj küçük şekil bozukluklarının önemsenmemesidir. Örneğin italik bir 'I' harfi ile düz 'I' harfleri sırf pikseller karşılaştırılacak olsa bu iki harf birbirine hiç benzemez, fakat şekli 40 kutuya bölen bir şekil içerik yapısı kullanılarak karşılaştırılırsa bu harfe ait iki şekil birbirine benzer olarak bulunur. Başka bir deyişle sırf piksellere bakarak karşılaştırılırken 20x20 = 400 piksel tek tek birbiri ile aynı olması beklenirken, şekil içerik yapısında 40 kutunun piksel sayılarının birbirine yakın olması beklenir. Böylece bir yaklaşıklık kullanılmış olur. Bakış açısı farklılığı, küçük dönüşler gibi şekil değişimlerinde oluşan küçük hatalar, belirli bir yaklaşıklık kullanılan bu yöntem ile ortadan kaybolur.

1.4. Çapraz İlinti

İki veri dizisinin benzerlik değerlerinin hesaplanmasında kullanılan çapraz ilinti (cross correlation) yöntemi (0, 1) açık aralığında bir değer üretir. Bu değer 1'e yakın olduğunda iki veri dizisinin birbiriyle benzer olduğu sonucuna varılır. Bu çalışmada varsayılan benzerlik eşik değeri 0.8 olmakla birlikte kullanıcı arayüzü ile değiştirilebilmektedir. İki veri dizisinin birbiriyle olan benzerliğini hesaplamak için başka yöntemler de vardır ve en uygun yöntem çapraz ilinti yöntemidir. Örneğin iki veri dizisindeki elemanların farklarının mutlak değerlerinin toplamı, benzerliğin bulunmasında kullanılabilir. Fakat elde edilen bu fark değerinin hangi eşikten küçük olduğunda benzerlik kararı verilebileceği belirlenemez. İki veri dizisi de sıfır değerlerinden oluşuyor olabilir veya büyük değerler içeren iki veri dizisi birbirine benzer olduğu halde farkları çok büyük olabilir. Bununla beraber her durumda çapraz ilinti yöntemi benzerliği ifade edebilmektedir. Çapraz ilinti yöntemi kullanılmadan önce ön bir karar mekanizması oluşturulması doğru olur. İki veri dizisinin de toplamının sıfıra yakın olması durumunda doğrudan benzerlik kararı verilebilir.

Başka bir durum da bir veri dizisinin toplamı sıfıra yakınken diğeri sıfırdan yeterince büyükse benzer olmama kararı verilir. Bu şekilde hesaplama hızı artırılır. Çünkü çapraz ilinti hesabı önemli ölçüde zaman alan bir iştir.

Çapraz İlinti hesaplanması Denklem 1.3'te verilmiştir. Bu hesaplamada karşılaştırılacak olan iki veri dizisinin elemanları kullanılır.

$$Çİ = \frac{\sum_{i=0}^{39} Ş_{1_i} * Ş_{2_i}}{\left[\sum_{i=0}^{39} Ş_{1_i}^{\ 2} * \sum_{i=0}^{39} Ş_{2_i}^{\ 2}\right]^{1/2}} \qquad (1.3)$$

Burada $Ş_1$ ve $Ş_2$, karşılaştırılacak olan şekil içerik yapılarıdır. Bu iki yapı da 40 elemanlı dizidir. Çİ ise çapraz ilinti değeridir.

Şekil içerik yapısındaki dizinin her bir elemanı, şekil içerik maskesindeki her bir kutuya düşen ayrıt pikseli sayısını verir. Bu iki veri dizisinin karşılaştırılmasında çapraz ilinti yöntemi kullanılır. Birinci şekil içerik yapısındaki kutuların karelerinin toplamı T_Kare1 ve diğer şekil içerik yapısındaki kutuların karelerinin toplamı ise T_Kare2, iki şekil içerik yapısındaki kutuların karşılıklı çarpımlarının toplamı da T_Çapraz_Çarpım olsun. Çapraz ilinti değeri Çapraz_İlinti = T_Çapraz_Çarpım / $(T\_Kare1 * T\_Kare2)^{1/2}$ olarak hesaplanır [7]. Daha önce de belirtildiği gibi bu hesabın sonucunda (0, 1) açık aralığında bir değer elde edilir ve 1'e yaklaşıldıkça benzerlik artar. Bu çalışmada varsayılan benzerlik eşiği değeri 0.8 olarak alınmıştır.

2. GEÇMİŞ ÇALIŞMALAR

2.1. Yapılan Geçmiş Çalışmalar

Mozaik çıkarma amaçlı geçmiş çalışmalarda alan tabanlı ve özellik tabanlı olarak iki yol izlenmiştir. Performansı ve zaman kullanımı alan tabanlı yöntemlere göre oldukça iyidir. Alan tabanlı yöntemlerin fazla zaman alması tüm resimlerdeki bütün piksellerle ilgilenmesinden kaynaklanır.

Mozaik çıkarma amaçlı yapılmış bir çalışmada fotoğraflar ikili olarak gruplanarak birbirine olan kaymalar hesaplanmıştır[1]. Bahsedilen çalışmada iki resmin birbirine olan kayıklığı, iki resim içerisinde pencere ilerleterek bu pencerelerin çapraz ilinti değerlerinin en çok desteklediği kayıklığa göre hesaplanmıştır. Bu raporda önerilen çözümde ise farklı olarak şekil içeriği yapısı ve bu şekil içeriği yapısının üzerinde tanımlı olduğu bir ayrıt haritası kullanılmıştır. Böylece arama penceresinin içeriği sıkıştırılarak arama hızı artırılmıştır. Ayrıt haritası çıkarılması için Canny Ayrıt Saptayıcısı kullanılmıştır[2,3]. Elde edilen ayrıt haritası Şekil İçeriği yapılarına dönüştürülür[4-6]. İki Şekil İçeriği yapısının karşılaştırılması için Çapraz İlgileşim yöntemi kullanılmıştır[7].

Otomatik panorama oluşturma amaçlı bir çalışmada SIFT dönüşümü kullanılmıştır[8]. Ölçekten bağımsız olarak tespit edilen özellik noktalarının kullanıldığı bu yöntemde rotasyon, ölçekleme ve belirli miktarda uzaysal deformasyonlar hesaba katılmıştır. Bundan dolayı bu yöntemde resimler üzerinde düzeltmeler yapılarak kaynaştırılır.

Diğer yöntemlerden oldukça farklı bir teknik kullanan başka bir mozaik çalışmasında morfolojik özellikler kullanılmıştır. Bu yöntemde resimdeki göl, nehir, ev çatısı gibi belirli şekiller ortaya çıkarılır. Bu şekiller iki resmin birbiriyle kaynaştırılması için kullanılır[9].

3. ÖNERİLEN ÇÖZÜM

3.1. Tanımı

Sıralı olarak verilmiş olan resimlerin otomatik olarak adreslenerek tek resimde birleştirilmesi, resim çiftlerinde örtüşen alanların bulunması ile gerçekleştirilir. İlk resimden başlanarak her bir resmin bir sonra gelen resimle örtüşen kısmı bulunur. Başka bir deyişle satır ve sütun eksenlerinde iki resmin birbirine olan kayıklığı belirlenir. Bir zincir şeklinde her resmin bir sonrakine göre kayma miktarı kullanılarak tüm resimlerin ilk resme göre kayıklıkları hesaplanır ve her bir resim evrensel bir resim içerisinde adreslenir. Son olarak tüm resimler tek bir resim içerisindeki adresine kopyalanır.

3.2. Algoritma

İki resimde örtüşen kısımların bulunması için bir kayıklığı destekleyen benzer pencerelerin sayısına ihtiyaç vardır. İki resim içerisinde iki piksellik adımlarla ilerlenerek 20x20 piksellik pencereler birbirleri ile karşılaştırılır. Bu benzerlikler bir listede tutulur. Bu listenin fazla yer kaplamaması için birbirine yakın olan benzerliklerden sadece biri seçilir. Elde edilen bu benzerlik listesinde aynı kayıklığı destekleyen benzerliklerin sayısı hesaplanır ve en çok desteklenmiş olan kayıklık, iki resmin birbirine olan kayıklığıdır.

Resim çiftlerinde ilerleyen pencerelerin karşılaştırılmasında şekil içerik yapısı kullanılır. Arama penceresinin şekil içerik yapısına göre parçalanması dilimler ve halkalar şeklinde gerçekleşir. Bu çalışmada 2 halka ve 20 dilim olmak üzere 40 kutulu bir şekil içerik yapısı kullanılmıştır.

Şekil içerik yapısı, ayrıt haritası çıkarılmış bir resim üzerinde geçerlidir. Ayrıt haritası, resimdeki gri değerlerin ani değiştiği çizgilerin işaretlendiği ikili resimlerdir. Ayrıt haritasında ayrıtların geçtiği piksellerin değeri “1”, diğer piksellerin değeri “0” olarak verilir. Resimlerin ayrıt haritasının çıkarılmasında Canny Ayrıt Saptayıcısı kullanılmıştır. Oluşturulan ayrıt haritasındaki ayrıtlarda genellikle hata oluşur.

Bunların inceltilmesiyle resim içindeki şekillerin iskeleti çıkarılabilir. Bununla beraber iskelet çıkarma işlemi de bu çalışmada denenmiş ve çok hataya neden olduğu gözlenmiştir. Bu durum şöyle açıklanabilir: Bu resim ulama işlemi zaten resimlerde ayrıtların belirli bir hatayla saptandığı varsayımıyla geliştirilmektedir. İskelet çıkarma aşamasından sonra hatasız ayrıtların silinip sadece hatalı ayrıtların kalmayacağı baştan kestirilememektedir. Dolayısıyla daha da hatalı bir durum ortaya çıkabilmektedir.

İki resmin birleştirilmesi için bir algoritma geliştirilmiştir. Bu algoritma daha sonra tüm resimlerin ikili olarak birleştirilmesiyle ikiden fazla resmin birleştirilmesine dönüştürülür. İki resmin büyük resimdeki yerlerinin belirlenmesi ve tek resimde birleştirilmesi algoritması aşağıda listelenmiştir.

- İki resimdeki her bir pikseli (0,0) kabul eden 20x20'lik pencereler için şekil içerik yapısı hesaplanarak Şekil İçerik Yapıları tablosunda tutulur.(2 adet tablo)
- Bu şekil içerik tablolarındaki her bir şekil içerik yapısının kutularının toplamı Pencere Yoğunluk Tablosunda tutulur. (2 adet tablo)
- Bu şekil içerik tablolarındaki her bir şekil içerik yapısının kutularının karelerinin toplamı Pencerelerin Karelerinin Toplamı tablosunda tutulur.
- İki resmin satır ve sütunlarında adım miktarı kadar ilerleyerek;
- İki resimde üzerinde bulunulan pikseli (0,0) kabul eden 20x20'lik pencerenin toplamları Pencere Yoğunluk Tablolarından alınır. Birinci pencerenin yoğunluğu 5'ten küçük ve ikinci pencerenin yoğunluğu 20'den büyükse veya tersi ise başka hesap yapılmadan "benzer değil" kararı verilir. Yoksa iki şekil içeriğinin çapraz ilinti hesaplanır. Çapraz ilinti 1.4'te anlatılmıştır. Çapraz ilinti hesaplanabilmesi için gerekli olan şekil içerik yapıları Şekil İçerik Yapıları tablosundan, şekil içerik yapılarındaki kutuların karelerinin toplamı ise Pencerelerin Karelerinin Toplamı tablosundan alınır.
- Çapraz ilinti değeri eşik değerinden büyükse bu benzerlik noktası benzerlik listesine eklenir.

- Benzerlik listesindeki benzerlik noktalarının 8-komşuluğundaki pencerelerin de benzerliklerine bakılır. Eğer biri bile benzer değilse bu benzerlik noktasının geçerliliği "Geçersiz" atanır.
- Benzerlik listesindeki bu benzerlik elemanları, iki resmin birbirine göre bir kayıklıkta olduğunu bildirir. İki resmin kayıklığının yaklaşık aynı olduğu durumdaki benzerlik elemanları sayılarak her bir benzerliğe frekans değeri atanır. Aynı kayıklığı destekleyen benzerlik elemanlarından sadece biri listede kalır.
- En yüksek frekanstaki benzerliğe göre iki resim birbirine göre kaydırılarak tek resimde birleştirilir.

İki resmi birleştiren algoritmada görüldüğü gibi iki resim arasındaki benzer pencereler bir tabloda tutulmaktadır. Buna **Tablo 3.1**'de bir örnek verilmiştir.

Tablo 3.1: İki resimdeki benzer noktaların tutulduğu liste görülmektedir.

Benzerlik indisi	Birinci resimdeki konum (satır,sütun)	İkinci resimdeki konum (satır,sütun)	Benzerlik değeri	Benzerlik frekansı	Benzerlik geçerliliği
0	56, 75	67, 155	0.92	1356	Geçerli
1	125, 43	156, 253	0.91	2	Geçerli
3	124, 43	154, 251	0.80	2	Geçersiz

Sadece iki resmi birleştiren uygulama birinci sürüm ve ikiden fazla resimleri birleştiren uygulama ikinci sürüm olarak adlandırılmış ve ekran görüntüsü bir sonraki bölümde verilmiştir. Bu iki uygulamanın arayüzü yaklaşık aynı olduğu için tek görüntü verilmiştir.

İkiden fazla resmin birleştirilmesi için geliştirilen algoritma adımları aşağıda listelenmiştir.

- Resimler sıralı olarak ikili gruplar halinde yukarıda verilen "iki resmi birleştiren" algoritma kullanılarak birbirine göre kayıklıkları hesaplanır.
- Tüm resimlerin ilk resme olan kayıklığı hesaplanır.

- En soldaki ve en sağdaki resim hesaplanır. En üstteki ve en alttaki resim hesaplanır.
- Evrensel resmin genişliği ve yüksekliği hesaplanır.
- Her bir resmin evrensel resimdeki yeri bulunur.
- Tüm resimler evrensel resimdeki yerine kaydedilir.

Bu çalışmadaki algoritmalardan da görüldüğü gibi var olan temel görüntü işleme yöntemleri kullanılarak özgün bir mozaik oluşturma yöntemi geliştirilmiştir.

4. UYGULAMA

4.1. Amaç

Önerilen resim adresleme ve resim mozaikleştirme yönteminin gerçeklenmesi ve bu yöntemin değişik test resimleri ile çalıştırılması amacıyla bir uygulama geliştirilmiştir. Uygulama geliştirme aracı olarak kullanıcı arayüzü ve grafik işlemlerinden soyutlanmak ve sadece algoritmaya odaklanmak amacıyla Visual Studio.NET C++ ortamı seçilmiştir. Kullanıcı arayüzünde kullanım kolaylığı göz önünde bulundurulmuştur. Uygulama aracılığıyla seçilen klasördeki tüm resim dosyalarının birleştirilmesi için sadece tek tuşa basmak yeterlidir. Bu işlemin ürünü tüm resimleri içeren ve "EvrenselResim.jpg" olarak isimlendirilen bir resim dosyasıdır. Bu dosya, birleştirilen dosyaların bulunduğu klasöre kaydedilir. Tahmin edileceği gibi bu isimdeki bir sonraki dosya birleştirme işlemine tekrar alınmamaktadır.

4.2. Yöntemin Tanıtımı

Bu uygulamada basitlik açısından kısıtlamalar konulmuştur. Bu kısıtlamalar, resimlerin boyutlarının 320x240 olarak kabul edilmesi ve resimlerin sıralı bir şekilde çekilmiş olduğudur. Örneğin, 1024x768 olarak çekilmiş olan resimler %31,25 oranında küçültülerek veya %31 oranında küçültüldükten sonra kalan kısmı kırpılarak 320x240 ebadına getirilmelidir. Resimlerin sıralı olması şartı, sırasız resimlerin birleştirilmesinin gerçeklenmemesinden dolayı konulmuştur. Sırasız verilen resimlerde bütün resimlerin ikili olarak zincir şeklinde eşleştirilmesi gerekmektedir. Bunun için resimlerin 32x24 ebadına küçültülmesi ve hızlı bir şekilde eşleştirilmesi düşünülmüş ancak zaman yetmediği için bu çalışma ertelenmiştir. Bundan dolayı resimler klasöre her zaman sıralı olarak kaydedilmelidir. Bu kısıt, uçaktan çekilen resimlerin her zaman sıralı olduğu varsayımına ters düşmemektedir. Çünkü Hava Aracı hareket ederken aynı zamanda 10 resim/s sıklıkta resim

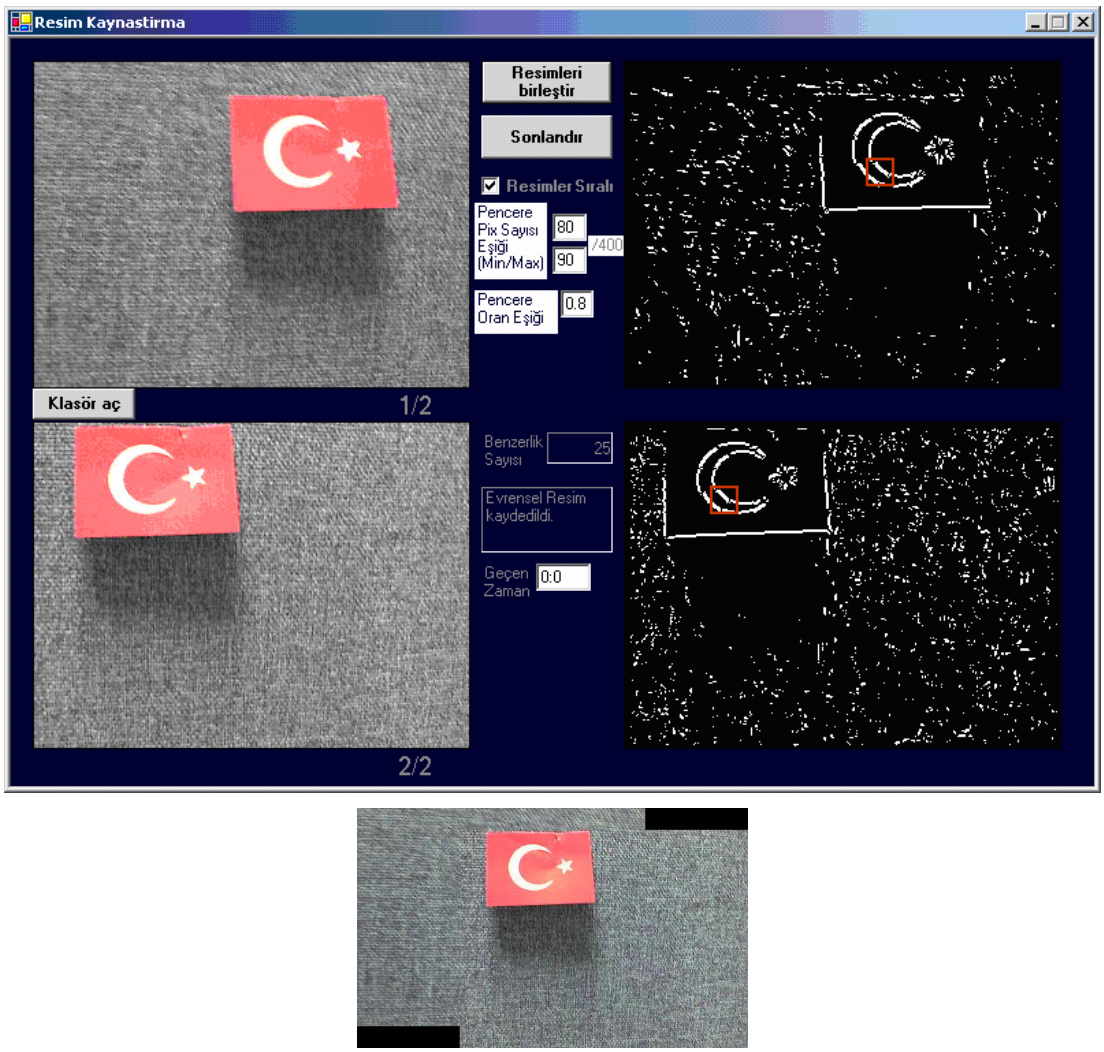
çekmektedir. Hava Aracının çektiği bir resim, bir sonraki ve bir önceki ile eşleşmektedir.

Kullanıcı arayüzünde verilen resim kümesi için değiştirilebilen parametreler bulunmaktadır. Bu parametreler benzer alanların aranması sırasında iki resimde ilerleyen arama penceresi ile ilgilidir. Arama penceresinin ebadı 20x20 pikseldir. Bundan dolayı bir arama penceresinin yoğunluğu en az "0", en fazla 400 olabilir. Resmin sadece ayrıt haritasına indirgenmesi, arama penceresinin kenar yoğunluğunu daha da düşürmektedir. Elde edilen sonuçlara göre bir arama penceresinin yoğunluğu 20 ile 140 arasında değişmektedir. Birleştirilmek üzere verilen bir resim gurubu için arama penceresini karakterize edebilecek minimum aralık belirlenerek alt ve üst eşik değerleri kullanıcı arayüzündeki kutulara yazılmalıdır. Bu yoğunluk aralığı ne kadar geniş seçilirse bulunan benzerlik sayısı o kadar çok olur fakat bu işlem oldukça fazla zaman alır, buna karşın bu aralık dar seçilirse bulunan benzerlik sayısı az ve geçen süre oldukça kısa olur. Bu parametrelerin en uygun şekilde seçilmesi gerekir. Sonuçlar bölümünde yapılan testlerde bu eşik değerleri 80–90 olarak seçilmiştir. Diğer parametre ise iki resimde ilerleyen pencerelerin oran eşiğidir. Başka bir deyişle iki pencerenin benzer olup olmadığının belirlenmesi için pahalı işlemler yapmadan önce bu iki pencerenin yoğunluklarının oranlarının 1'e ne kadar yakın olduğu hesaplanır. Bu da resimlerin birbirinden ne kadar farklılaştığı ile ilgilidir. Genel olarak bu parametre için 0,8 değeri en uygundur ve Sonuçlar bölümünde yapılan testlerde hep 0,8 olarak kullanılmıştır. Burada bahsedilen en uygun değerler Tablo 4. 1'de görülmektedir.

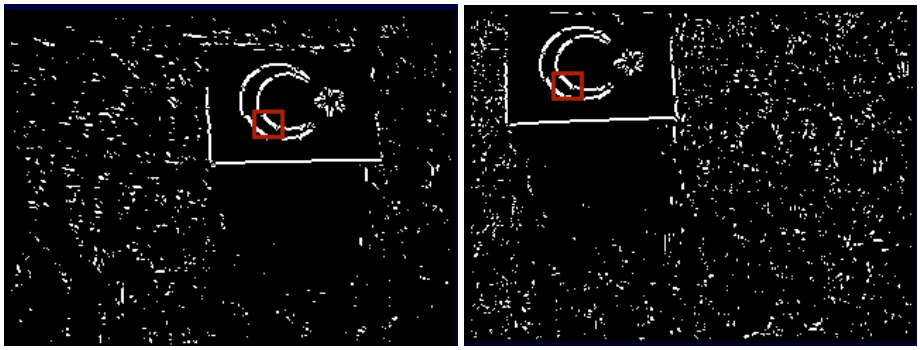
Tablo 4. 1: Kullanıcı arayüzündeki parametrelerin değer aralıkları ve en uygun değerleri. (Penc.boyu=20)

Parametreler	Değer aralığı	En uygun değer
Pencere yoğunluk alt eşiği	40 – 140	80
Pencere yoğunluk üst eşiği	40 – 140	90
Pencereler arası oran eşiği	0,0 – 1,0	0,8

Kullanıcı Arayüzü Şekil 4. 1'de gösterilmiştir. Ayrıca hem Şekil 4. 1'de hem de Şekil 4. 2'de Kullanıcı Arayüzünde bir adresleme işlemi sonrası bulunan benzer pencereler iki kırmızı kare şeklinde ayrıt haritalarının üzerinde görülmektedir



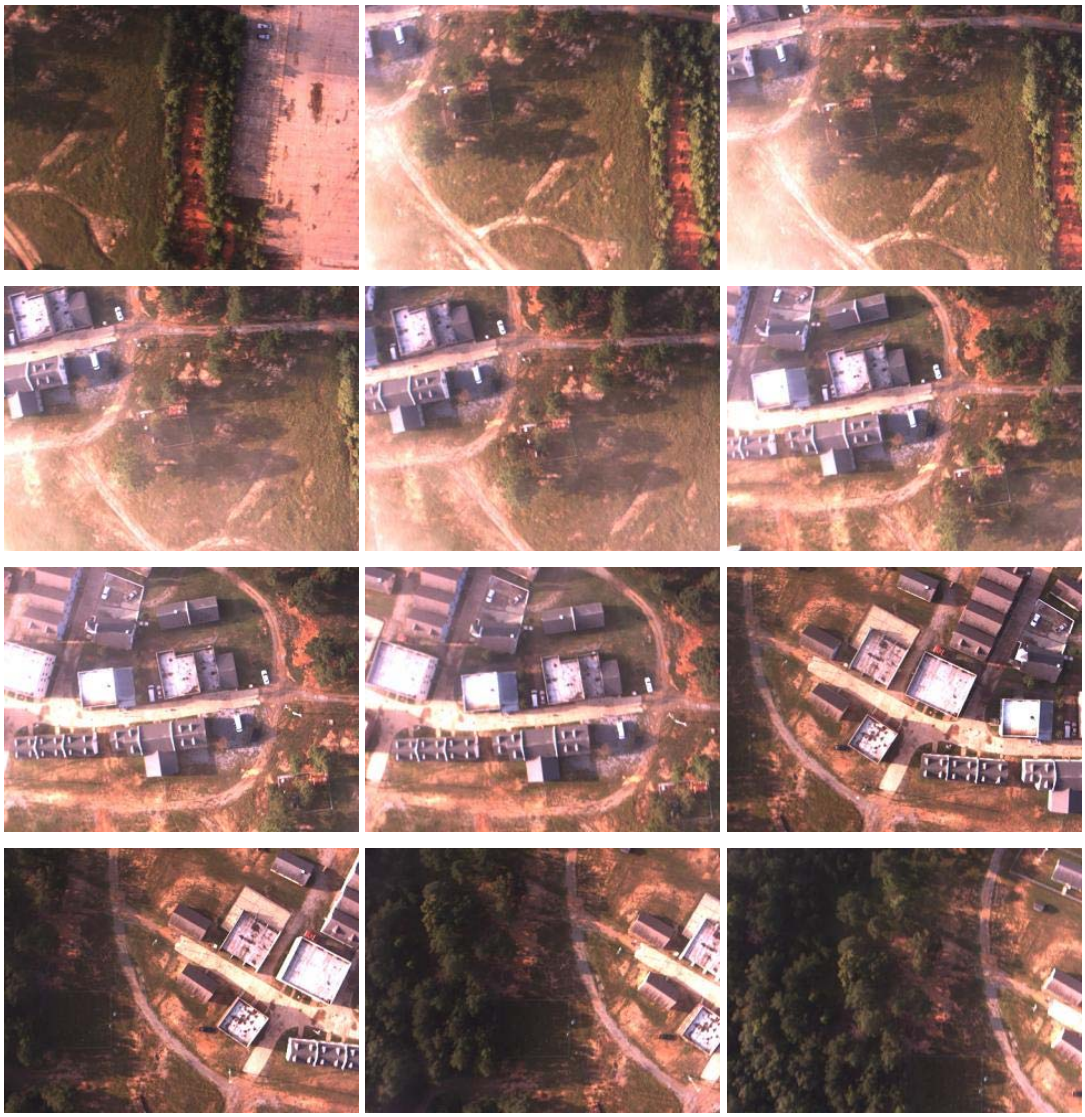
Şekil 4. 1: Tablo 4.1'deki en uygun değerlerle iki resmin mozaik yapılması (Süre 1s)



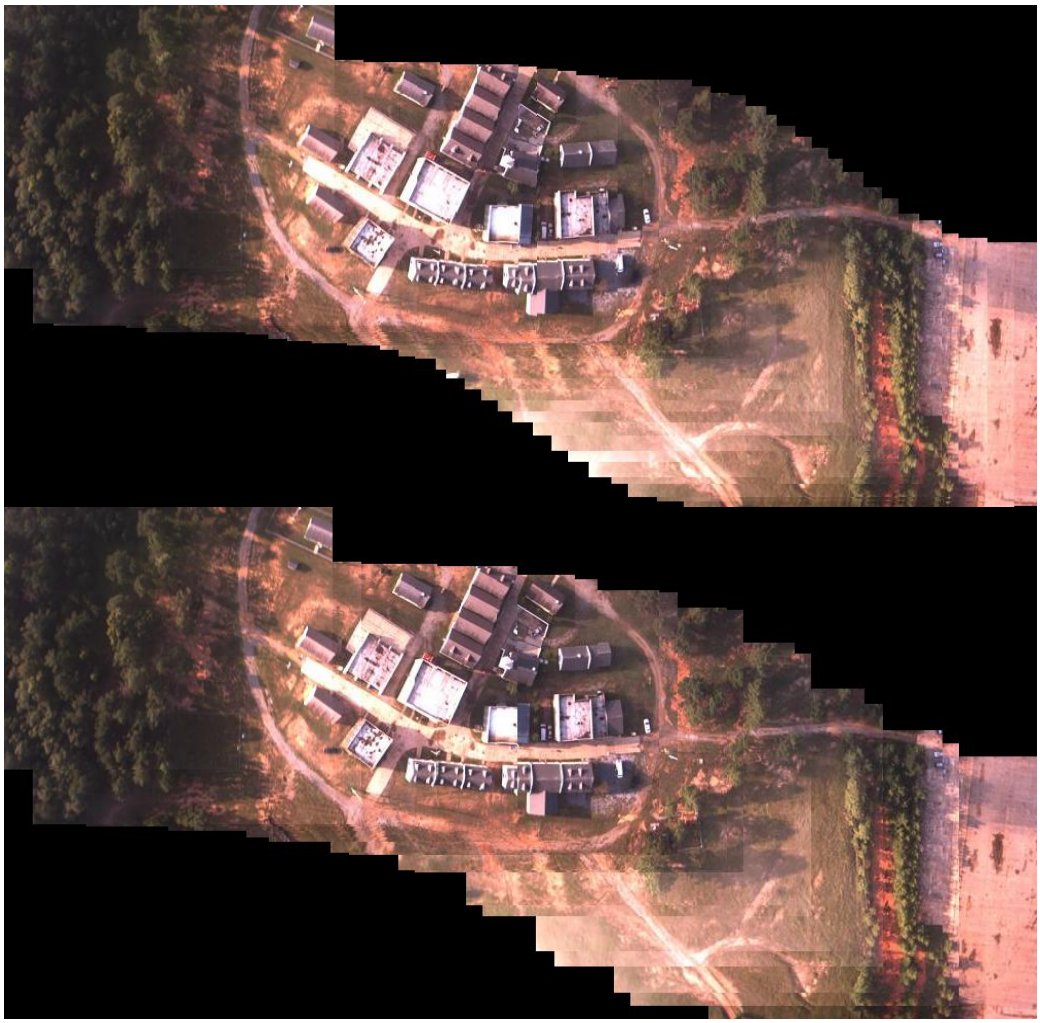
Şekil 4. 2: Mozaik uygulaması resimleri ikili gruplar halinde işlemektedir. İki resmin ayrıt haritası ve arama penceresi gösterilir.

5. SONUÇ VE TARTIŞMA

Bu çalışmada elde edilen yöntemin test edilmesi için birçok mekânın fotoğrafları çekilmiş ve mozaiği çıkarılmıştır. Bu bölümdeki şekillerde bu mekânlarda çekilen fotoğraflar ve bunların mozaikleştirilmesiyle elde edilen resimler verilmiştir. Bu testlerden elde edilen sonuçlara göre bu yöntemin uzaktan çekilen fotoğraflarda yakından çekilen fotoğraflara göre daha düzgün çalıştığı anlaşılmıştır. Uzaktan çekilen fotoğraflarda şekil bozulmaları olmamaktadır. Bunun en belirgin örneği Şekil 5. 10'deki balkonun kırıklı ve uzaktaki manzaranın düzgün kaynaştırılmış görüntüsüdür. Ayrıca uçaktan çekilen fotoğraflarda şekil bozulması yok denilecek kadar azdır. Bu uygulamanın test edilmesinde kullanılan fotoğrafların bir kısmı Baykar Makine şirketinde hava aracı ile çekilmiştir. Test verisi olarak Şekil 5. 1'de mozaik uygulaması programına adreslenmek üzere verilen 37 resimden bazıları görülmektedir. Bu resimler 150 m yükseklikten 10 resim/s frekansla çekilmiştir. Bu 37 resmin tek resimde adreslenmiş hali Şekil 5. 2a'da, ve bu resimler her dört resimden ikisi atılarak uygulamaya verildiğinde oluşan mozaik resim Şekil 5. 2b'de görülmektedir. Aradan resim atıldığında büyük yaklaşıklıkla aynı resim elde edilmiştir. Bu resimlerin adresleme süresi yaklaşık 1.5 dakikadır.

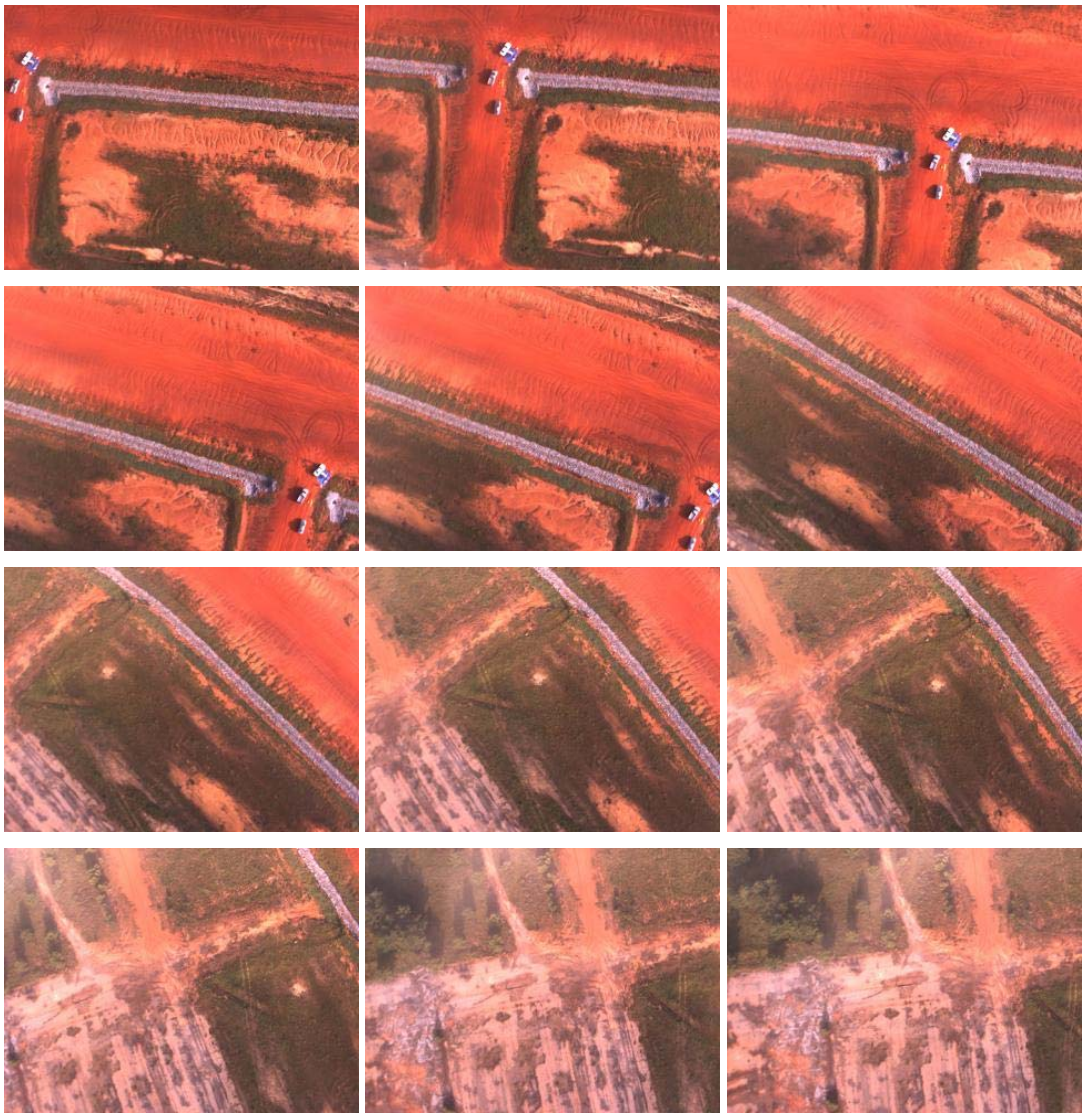


Şekil 5. 1: Mozaik uygulama programına adreslenmek üzere verilen 37 resimden bazıları görülmektedir.

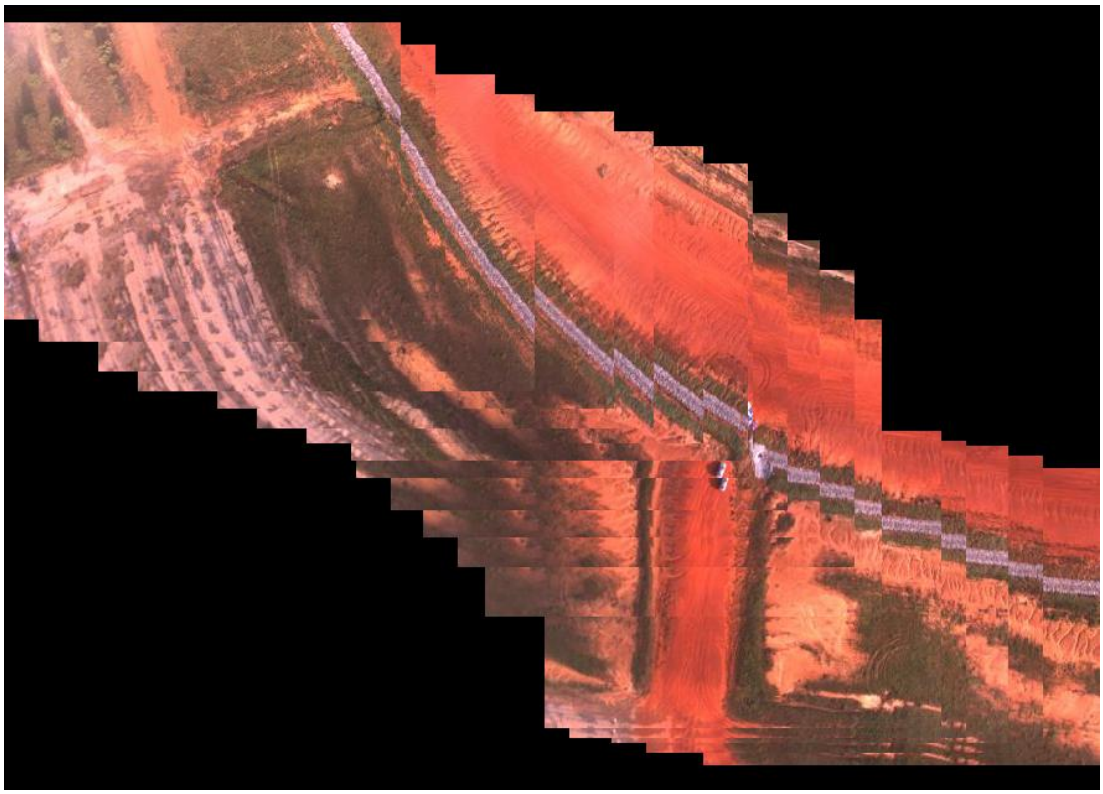


Şekil 5. 2: Şekil 5.1'de bahsedilen resimlerle oluşturulmuş iki farklı mozaik resim görülmektedir.

Bu çalışmada döndürülmüş resimler işlenmemekle birlikte, döndürülmüş resimler için de test yapılmıştır. Bu amaçla yapılan test resimleri ve sonucu Şekil 5. 3 ve 'te görülmektedir. Bu resimler hava aracının yaklaşık 90˚ sola döndüğü bir zamanda çekilmiştir. Hava aracının dönerek çekmesi ve bu resimlerin adreslenmesinde açı farkının hesaba katılmamasından dolayı adresleme tam olarak gerçekleştirilememiştir. Oluşan mozaik resmi Şekil 5. 4'te görülmektedir. Adresleme süresi yaklaşık 1 dakikadır.

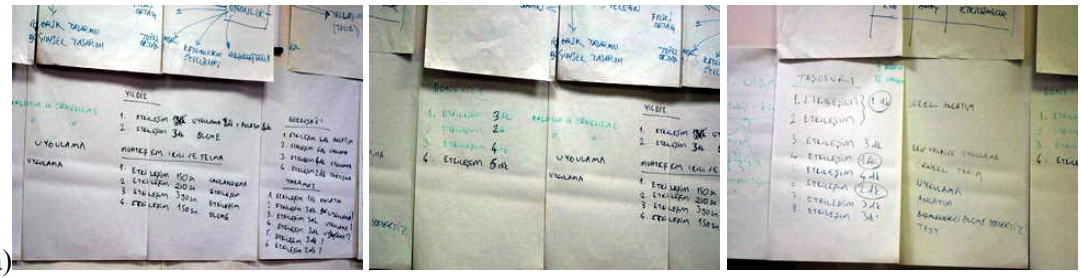


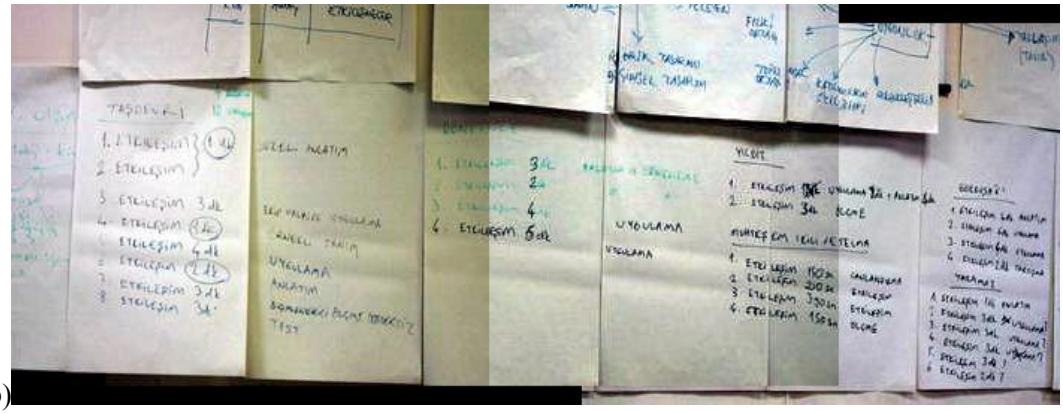
Şekil 5. 3: Mozaik uygulama programına adreslenmek üzere verilen 21 resimden bazıları görülmektedir.



Şekil 5. 4: Şekil 5.3'te bahsedilen fotoğraflarla yapılan mozaik resim görülmektedir.

Yakından çekilen ve perspektif değişikliği önemsiz olan resimlerle de test yapılmıştır. Şekil 5. 5, Şekil 5. 6 ve Şekil 5. 7 buna örnek olarak verilmişlerdir. Adresleme süreleri birincisinde 20 s, ikincisinde 1 s ve üçüncüsünde 4 s olarak ölçülmüştür.

a) 

b) 

Şekil 5. 5: Şekilde bir ders notunu içeren kâğıtların fotoğrafları ve bunların mozaik resmi görülmektedir.

Bakış açısı farklılığı olmadan yakından çekilerek elde edilmiş fotoğraflar üzerinde yapılmış bir mozaik uygulaması testi Şekil 5. 6'da verilmiştir.

Şekil 5. 6: Şekilde iki halı parçası resmi ve bunların mozaik sonucu oluşan evrensel resim görülmektedir.

Şekil 5. 7: Şekilde iki oda zemini resmi ve bunların tek resimde adreslenmesiyle oluşan evrensel resim görülmektedir.

Bu algoritmada desteklenmemekle birlikte yakından çekilen ve belirli oranda perspektif değişimi olan fotoğraflarla da testler yapılmıştır. Şekil 5. 8 buna örnek olarak verilmiştir. Mozaik yapma süresi 1 s'dir. Oluşan şekil değişimleri düzeltilmemektedir.

Şekil 5. 8: Şekilde bir ofiste çekilen üç fotoğraf ve bunların tek resimde adreslenmesiyle oluşan evrensel resim görülmektedir.

Bu algoritmada resimlerin birbirine olan kaymasını bulurken bir oylama mantığı kullandığı için fotoğraflarda hareketli olan kısımlar olsa bile mozaik yapabilmektedir. Buna örnek olarak Şekil 5. 9 verilmiştir. Bu iki fotoğrafta televizyondaki görüntüler birbirinden farklıdır. Genel olarak bir kayıklığın oylaması en fazla olduğu için bu mozaik işlemi televizyondaki hareketten etkilenmemiştir.

Şekil 5. 9: Şekilde bir salonda çekilen iki fotoğraf ve bunların mozaik resmi görülmektedir.

Uzaktan çekilen fotoğrafların mozaik çıkarılmasıyla ilgili bir test olarak bir evin balkonundan çekilen manzara fotoğraflarıyla mozaik yapılmıştır. Bu testte kullanılan resimlerden bazıları ve bu resimlerin tamamını içeren mozaik resmi Şekil 5. 10'da görülmektedir. Bu resimde balkon mermerlerinin kırıklı görünmesi, yakındaki nesnelerin perspektiften daha çok etkilenmesiyle açıklanabilir. Bununla birlikte balkonun dışındaki uzak manzara görüntüleri doğru olarak birleştirilmiştir. Bu resimlerin adresleme süresi 28 s olarak ölçülmüştür.

Şekil 5. 10: Şekilde bir evin balkonundan çekilen 16 fotoğraftan bazıları ve bu 16 fotoğrafın tek resimde adreslenmesiyle oluşan evrensel resim görülmektedir.

Algoritmanın test edilmesi sırasında Şekil 5. 1'de bir kısmı verilen resimler üzerinde algoritmadaki parametreler değiştirilerek testler yapılmıştır. Değiştirilerek test yapılan parametreler şunlardır: resim boyutu, Canny ayrıt saptayıcısındaki Gaussian maskesinin varyansı ve arama penceresinin boyutu. Resim boyutu 320x240, 160x120 ve 80x60 olarak değiştirilerek test edilmiştir. Resim boyutu küçüldükçe ayrıntı azaldığı için başarım düşmekte, fakat çalışma hızı büyük oranda artmaktadır. Gaussian maskesinin varyansı 0.5'ten 2.0'a kadar belirli aralıklarla artırılarak testler yapılmıştır. Bu değer 0.5 olduğunda ayrıntılar artmakta, ayrıt olmayan yerler de ayrıt olarak görülmekte ve başarım düşmektedir. Varyans değeri 2.0'ye yaklaşırken sadece ayrıt olan yerler ayrıt olarak işaretlenmekte ve ayrıntı azalmaktadır, aynı zamanda başarımın ve hızın arttığı gözlenmiştir. Arama penceresinin boyutu 6x6, 10x10, 16x16, 20x20 ve 24x24 olarak değiştirilerek her biri için testler yapılmış ve en uygun boyutun 16x16 olduğu gözlenmiştir. Test sonuçlarının verildiği tablolarda, mozaik sonucunun gözle ölçülmesinden dolayı mozaik başarımı sözle ifade edilmiştir. Tüm resimler yaklaşık olarak doğru yerine yerleştirilmesi durumunda mozaik başarımı "Düzgün", aksi durumda "Bozuk" olarak ifade edilmiştir. Resimlerin yerlerindeki hata toplamı 100x100 pikselden fazla ise "Toplam hata fazla", aksi durumda "Toplam hata önemsenmez" olarak ifade edilmiştir.

Pencere boyutu 6x6, Gaussian varyansı 1.4 ve resim boyutu 320x240 olarak seçildiğinde doğru mozaik oluşturulamamıştır. Bu testte pencere ayrıt piksel sayısı alt/üst eşiklerin değişimi ve buna bağlı olarak çalışma süreleri Tablo 5. 1'de verilmiştir.

Tablo 5. 1: Pencere boyu 6, varyans 1.4 ve resim boyutu 320x240 seçilerek yapılmış test sonuçları görülmektedir.

Pencere ayrıt piksel sayısı alt/üst eşikleri	7-12	7-15	10-15	15-25
Çalışma süresi	4.5 dk	6.5 dk	1 dk	1 s
Mozaik Başarımı	Bozuk	Bozuk	Bozuk	Bozuk

Pencere boyutu 10x10, Gaussian varyansı 1.4 ve resim boyutu 320x240 olarak seçildiğinde doğru mozaik oluşturulmuştur. Bu testte pencere ayrıt piksel sayısı alt/üst eşiklerin değişimi ve buna bağlı olarak çalışma süreleri Tablo 5. 1'de verilmiştir.

Tablo 5. 2: Pencere boyu 10, varyans 1.4 ve resim boyutu 320x240 seçilerek yapılmış test sonuçları görülmektedir.

Pencere ayrıt piksel sayısı alt/üst eşikleri	Çalışma süresi	Mozaik Başarımı
18-24	4 dk	Toplam hata fazla, Düzgün
20-24	2 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
22-28	2 dk	Toplam hata fazla, Bozuk
22-30	4 dk	Toplam hata önemsenmez, Düzgün
24-28	0.75 dk	Toplam hata fazla, Bozuk
24-30	1.7 dk	Toplam hata fazla, Bozuk
25-30	1 dk	Toplam hata fazla, Bozuk
20-27	3 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
20-28	4 dk	Toplam hata önemsenmez, Düzgün
20-30	4.5 dk	Toplam hata önemsenmez, Düzgün

Pencere boyutu 16x16, Gaussian varyansı 1.4 ve resim boyutu 320x240 olarak seçildiğinde doğru mozaik oluşturulmuştur. Bu testte pencere ayrıt piksel sayısı alt/üst eşiklerin değişimi ve buna bağlı olarak çalışma süreleri Tablo 5. 1'de verilmiştir.

Tablo 5. 3: Pencere boyu 16, varyans 1.4 ve resim boyutu 320x240 seçilerek yapılmış test sonuçları görülmektedir.

Pencere ayrıt piksel sayısı alt/üst eşikleri	Çalışma süresi	Mozaik Başarımı
45-50	0.5 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
45-55	2 dk	Toplam hata önemsenmez, Düzgün
45-60	4.5 dk	Toplam hata önemsenmez, Düzgün
50-60	2.5 dk	Toplam hata önemsenmez, Düzgün
55-65	2 dk	Toplam hata önemsenmez, Düzgün

Pencere boyutu 20x20, Gaussian varyansı 1.4 ve resim boyutu 320x240 olarak seçildiğinde doğru mozaik oluşturulmuştur. Bu testte pencere ayrıt piksel sayısı alt/üst eşiklerin değişimi ve buna bağlı olarak çalışma süreleri Tablo 5. 1'de verilmiştir.

Tablo 5. 4: Pencere boyu 20, varyans 1.4 ve resim boyutu 320x240 seçilerek yapılmış test sonuçları görülmektedir.

Pencere ayrıt piksel sayısı alt/üst eşikleri	Çalışma süresi	Mozaik Başarımı
40-50	2 dk	Toplam hata fazla, Bozuk
50-60	2.5 dk	Toplam hata önemsenmez, Düzgün
60-70	2.5 dk	Toplam hata önemsenmez, Düzgün
70-80	3 dk	Toplam hata önemsenmez, Düzgün
80-90	2 dk	Toplam hata önemsenmez, Düzgün
90-100	2 dk	Toplam hata önemsenmez, Düzgün
100-110	1.5 dk	Toplam hata fazla, Düzgün
110-120	0.5 dk	Toplam hata fazla, Bozuk

Hem varyans hem de pencere ayrıt piksel sayısı eşik değerleri değiştirilerek yapılan test sonuçları Tablo'da verilmiştir. Bu sırada Pencere boyutu 20x20 ve resim boyutu 320x240 olarak sabit tutulmuştur.

Tablo 5. 5: Pencere boyu 20, resim boyutu 320x240 durumunda varyans ve pencere ayrıt piksel eşikleri değiştirilerek yapılan test sonucu görülmektedir.

Penc. ayrıt say. eşikleri	Gaussian Varyans değeri	Çalışma süresi	Mozaik Başarımı
50-60	2	3 dk	Toplam hata fazla, Bozuk
60-70	2	2 dk	Toplam hata fazla, Bozuk
70-80	2	2 dk	Toplam hata önemsenmez, Düzgün
80-90	0.5	10 dk	Toplam hata fazla, Bozuk
80-90	1	3 dk	Toplam hata önemsenmez, Düzgün
80-90	2	2 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
90-100	1	3 dk	Toplam hata önemsenmez, Düzgün

Pencere boyutu 24x24 ve resim boyutu 320x240 olarak seçildiğinde doğru mozaik oluşturulmuştur. Bu testte pencere ayrıt piksel sayısı alt/üst eşiklerin değişimi, varyans değişimi ve buna bağlı olarak çalışma süreleri Tablo 5. 1'de verilmiştir.

Tablo 5. 6: Pencere boyu 24, resim boyutu 320x240 durumunda varyans ve pencere ayrıt piksel eşikleri değiştirilerek yapılan test sonucu görülmektedir.

Penc. ayrıt say. eşikleri	Gaussian Varyans değeri	Çalışma süresi	Mozaik Başarımı
80-90	1.4	1 dk	Toplam hata önemsenmez, Düzgün
90-100	1.4	1.5 dk	Toplam hata önemsenmez, Düzgün
90-100	2	1.5 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
90-105	1.4	4 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
90-110	1.4	7 dk	Toplam hata önemsenmez, Düzgün
100-110	1.4	1.5 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
100-110	2	2 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
105-115	1.4	1.5 dk	Toplam hata önemsenmez, Düzgün
110-120	1.4	2 dk	Toplam hata önemsenmez, Düzgün
110-120	2	1.5 dk	Toplam hata önemsenmez, Düzgün
112-118	1.4	1 dk	Toplam hata önemsenmez, Düzgün
112-128	1.4	5 dk	Toplam hata önemsenmez, Düzgün
115-120	1.4	0.5 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
120-130	1.4	2.5 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
120-130	2	2 dk	Toplam hata önemsenmez, Düzgün
130-140	1.4	2 dk	Toplam hata önemsenmez, Düzgün
130-140	2	2 dk	Toplam hata fazla, Bozuk
140-150	1.4	1.5 dk	Toplam hata fazla, Bozuk

Gaussian varyansı 2 ve resim boyutu 160x120 olarak seçildiğinde doğru mozaik oluşturulmuştur. Bu testte pencere ayrıt piksel sayısı alt/üst eşikleri ve pencere boyutu değiştirildiğinde başarım ve çalışma süreleri Tablo 5. 1'de verilmiştir.

Tablo 5. 7: Resim boyutu 160x120 ve varyans 2 seçilmesi durumunda pencere boyu ve pencere ayrıt piksel eşikleri değiştirilerek yapılan test sonucu görülmektedir.

Penc. ayrıt say. eşikleri	Pencere boyutu	Çalışma süresi	Mozaik Başarımı
20-30	10	0.5 dk	Toplam hata fazla, Bozuk
30-40	10	0.5 dk	Toplam hata fazla, Düzgün (Sadece bir resim kaymış)
30-40	16	0.1 dk	Toplam hata fazla, Bozuk
40-50	16	0.1 dk	Toplam hata önemsenmez, Düzgün
40-60	16	0.5 dk	Toplam hata önemsenmez, Düzgün
70-90	20	1 dk	Toplam hata fazla, Bozuk
80-90	20	0.5 dk	Toplam hata fazla, Bozuk

Resim boyutu 80x60 yapıldığında birçok parametre kombinasyonu ile testler yapılmış, ancak hiçbir şekilde düzgün mozaik elde edilememiştir. Bu durumda detay azaldığı için benzerlik bulma şansı azalmaktadır. Sadece pencere boyutu 24x24, pencere ayrıt piksel sayısı alt/üst eşikleri 0/570, varyans ise 2 olarak seçildiğinde 0.5 dakika süren bir işlem sonunda düzgün bir mozaik elde edilebilmiştir, ancak bunda bile bir fotoğraf kaymış olarak oluşmaktadır.

KAYNAKLAR

[1] Orduyilmaz, A., 2000. Automated Image Registration And Mosaicking For Multi-Sensor Images Acquired By A Miniature Unmanned Aerial Vehicle Platform. *Mississippi State University, A Thesis Submitted for the Degree of Master of Science, August 2006.*

[2] Kuntz, N., Canny Tutorial. 5 Mar. 2006. 1 Feb. 2007

[3] Green, B., Canny Edge Detection Tutorial. 1 Mar. 2002. 1 Feb. 2007

[4] Belongie, S., Malik, J., Puzicha, J., 2000. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No.24, April 2002.*

[5] Yusuf, M., Haider, T., 2004. Recognition of Handwritten Urdu Digits. *INMIC 2004*.

[6] Pillai, B., 2004. Shape Descriptor using Polar Plot for Shape Recognition. *Clemson University. Graduate paper for Digital Image Processing*.

[7] Lewis J. P., 1995. Fast Normalized Cross-Correlation. *Expanded Version of a Paper from Vision Interface, 1995.*

[8] Brown M., Lowe D. G., 2003. Recognising Panoramas. *ICCV 2003.*

[9] Pierre Soille, 2006. Morphological Image Compositing. *IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 28, No. 5, May 2006.*

ÖZGEÇMİŞ

Mesut PAK, 08/01/1981 tarihinde Kahramanmaraş'ta doğmuştur. Ortaöğrenimini Kahramanmaraş Açıköğretim Lisesi'nde tamamladıktan sonra İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü'nde lisans eğitimi almıştır. 05.2003-05.2004 tarihleri arasında 2U Information Technologies Şirketi'nde İnteraktif TV Uygulamaları Geliştiricisi olarak çalışmış, 05.2004-05.2005 tarihleri arasında Baykar Makina A.Ş'nin ARGE bölümünde Bilgisayar Mühendisi olarak çeşitli aviyonik yazılımlarında görev almıştır. Mayıs 2005'ten bu güne TÜBİTAK Marmara Araştırma Merkezi Bilişim Teknolojileri Enstitüsü'nde Araştırmacı olarak çalışmalarına devam etmektedir.