

Araç Destek Sistemleri İçin Kuş Bakışı Görüntü Dönüşümü

Turgut Doğan¹, Eser Sert², Deniz Taşkın³

^{1,3}Trakya Üniversitesi, Bilgisayar Mühendisliği Bölümü, Edirne

²Trakya Üniversitesi, Teknik Bilimler Meslek Yüksek Okulu, Edirne

¹turgutdogan@trakya.edu.tr, ²esersert@trakya.edu.tr, ³deniztaskin@trakya.edu.tr,

Özet: Resim ya da videolar üzerinde sayısal görüntü işleme ve analiz teknikleri kullanılarak çeşitli görüntü dönüşümleri yapılabilmektedir. Bu dönüşümlerin günlük hayata uyarlanmalarıyla çok zor görünen birçok görev kolaylıkla gerçekleştirilebilmektedir. Araçlarda kullanılan park destek sistemi de bu kolaylıklardan biridir. Araçın çeşitli yerlerinde bulunan kameraların algıladığı görüntüler, içeride bulunan kontrol ekranına kuş bakışı dönüşüm işlemi yapılarak ve birleştirilerek aktarılır. Bu çalışmada bu dönüşümlere örnek, OpenCV açık kaynak kodlu görüntü işleme kütüphanesi ile geliştirilmiş uygulamalar anlatılmaktadır. Ayrıca bu örnekler birleştirilerek görüntünün kuş bakışı dönüşümünü sağlayan bir uygulama geliştirilmiştir.

Anahtar Kelimeler: Görüntü İşleme, Görüntü Dönüşümü, Kuş Bakışı Görüntü Dönüşümü, OpenCV

Bird's Eye View Transformation For Vehicle Assistance Systems

Abstract: Image transforms can be performed on image or video files by using some analysis techniques and digital image processing. Several difficult tasks in daily life are performed easily by using these transforms. Parking Assistance System in vehicles is one of these facilities. Images, detected by cameras located at the different spots of a vehicle are shown in the control monitor after combining and applying bird's eye view transforming operation. In this study, samples of these transformations with OpenCV which is an open source image processing library is given. Also, an application which combines these samples to provide bird's-eye view transformation of a photo is developed.

Key Words: Image Processing, Image Transform, Bird's Eye View Transform, OpenCV

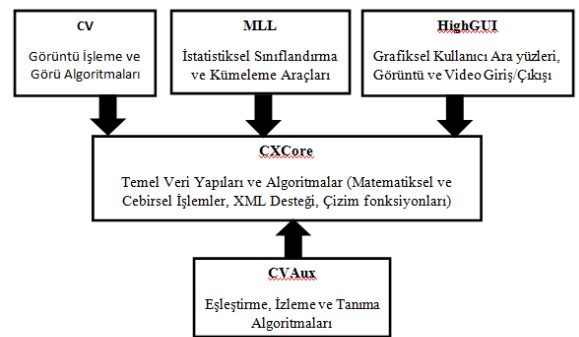
1. Giriş

Görüntü işlemeye dayalı olarak geliştirilen uygulamalar, günlük yaşamın vazgeçilmez birer parçası konumundadır. Araç çevrelerini görüntüleme sistemlerinde kullanılan kuş bakışı görüntü dönüşümü de bu uygulamalardan biridir. Bu çalışmada OpenCV açık kaynak kodlu görüntü işleme kütüphanesi ile geliştirilen dönüşümler kullanılarak kuş bakışı görüntü dönüşümü gerçekleştirilmiştir, böylece yatay ekseninde çekilmiş bir görüntünün seçilen 4 adet noktası arasında kalan kısımlarının tepeden çekilmiş formu elde edilmiştir.

2. OpenCV

OpenCV, bir resim ya da video içindeki anlamlı bilgileri çıkarıp işleyebilmek için INTEL tarafından C ve C++ dilleri kullanılarak geliştirilmiş, açık kaynak kodlu bir "Bilgisayarla Görme" kütüphanesidir

OpenCV kütüphanesi, beş temel bileşenden oluşmaktadır. Bu bileşenler Şekil 1'de görülmektedir.



Şekil 1. OpenCV Bileşenleri [1]

Computer Vision kelimesinin baş harfleri kullanılarak isimlendirilen CV bileşeni, temel resim işleme fonksiyonları ve Bilgisayarla Görü/Görme için kullanılan yüksek seviyeli algoritmaları bünyesinde barındıran beş temel kütüphaneden biridir. Machine Learning kelimesinin baş

harfleri kullanılarak isimlendirilen MLL bileşeni, adından da anlaşılacağı üzere Makine Öğrenmesi dalı için gerekli istatistiksel verilere ulaşmak, kütüphanesi içerisinde tanımlanmış pek çok nesneyi yaratabilmemizi sağlayan bir grafik arabirimi olmakla beraber, resim ve videoları kaydetmek, yüklemek, hafızadan silmek için gerekli giriş/çıkış (I/O) fonksiyonlarını da içerir [1].

CXCore bileşeni, OpenCV'ye ait IplImage, cvPoint, cvSize, cvMat, cvHistogram vs gibi veri yapılarını bünyesinde barındıran, XML desteği de sağlayan bir kütüphanedir. Son olarak CvAux bileşeni, şablon eşleştirme, şekil eşleştirme, bir objenin ana hatlarını bulma, yüz tanıma, ağız hareketleri izleme, vücut hareketlerini tanıma ve kamera kalibrasyonu gibi daha pek çok deneysel algoritmaları bünyesinde barındıran kütüphanedir.

Akademik ve ticari kullanımı ücretsiz olan OpenCV, Windows, Linux, MacOS X gibi farklı platformlarda kullanılabilir [2]. Intel'in görüntü işleme laboratuvarlarında geliştirilen ve hız açısından optimize edilen OpenCV kütüphanesi, gerçek zamanlı uygulamalar hedef alınarak geliştirilmiştir.

3. Temel Görüntü Dönüşümleri

3.1 Görüntü Eşikleme

Bu fonksiyon genellikle gri tonlu bir görüntüden ikili (binary) görüntü oluşturmak için kullanılır. Renkli görüntülerle de kullanılabilir. Kaynak görüntünün piksel değerlerini çok büyük ya da çok küçük değerlere filtreler. Çoğunlukla görüntülerdeki gürültüyü kaldırmak için kullanılır [2]. Herhangi bir görüntüdeki nesne, piksel değerlerinden yola çıkarak arka planından ayrılabilir. Şekil 2'de eşikleme tipleri gösterilmektedir.



$$\text{Hedef}(x,y) = \begin{cases} \text{maks. değer} & \text{eğer kaynak}(x,y) > E \\ 0 & \text{aksi takdirde} \end{cases}$$

a) Binary Tip



$$\text{Hedef}(x,y) = \begin{cases} 0 & \text{eğer kaynak}(x,y) > E \\ \text{maks. değer} & \text{aksi takdirde} \end{cases}$$

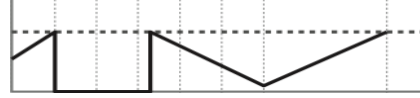
b) Binary Inverted Tip



mevcut verileri sınıflandırmak için kullanılan fonksiyonları içeren diğer bir kütüphanedir. HighGUI bileşeni, slider, form gibi OpenCV

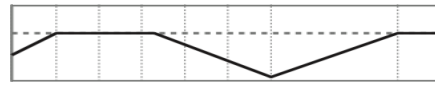
$$\text{Hedef}(x,y) = \begin{cases} \text{eşik değeri} & \text{eğer kaynak}(x,y) > E \\ \text{ kaynak}(x,y) & \text{aksi takdirde} \end{cases}$$

c) To Zero Tip



$$\text{Hedef}(x,y) = \begin{cases} \text{ kaynak}(x,y) & \text{eğer kaynak}(x,y) > E \\ 0 & \text{aksi takdirde} \end{cases}$$

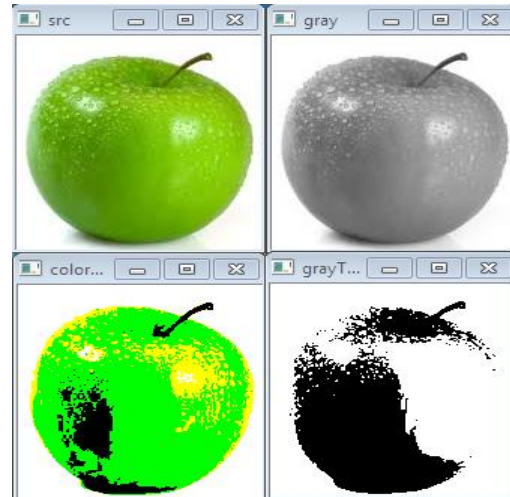
d) To Zero Inverted Tip



$$\text{Hedef}(x,y) = \begin{cases} 0 & \text{eğer kaynak}(x,y) > E \\ \text{ kaynak}(x,y) & \text{aksi takdirde} \end{cases}$$

e) Truncate Tip

Şekil 2. Eşikleme Tipleri ve Çalışma Biçimleri [2]



Şekil 3. Renkli ve Gri Görüntülerle Görüntü Eşikleme

3.2 Kenar Bulma

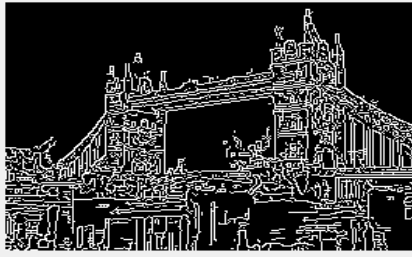
Bir kaynak görüntünün kenarlarını bulmak için kullanılan 3 yöntem mevcuttur. Bunlar canny, sober ve laplace yöntemleridir. Örneğin canny yönteminde, görüntünün kenarları bulunur, işaretlenir ve canny algoritması kullanılarak çıktı görüntüde gösterilir. 2 adet eşik değerini parametre olarak alır. Bunlardan büyük olanı görüntüdeki en büyük kenarın başlangıç bölümünü bulmak için kullanılır. Küçük olanı ise bu eşik değerleri arasında kalan kenarlar arasında bağlantı kurmak için kullanılır.



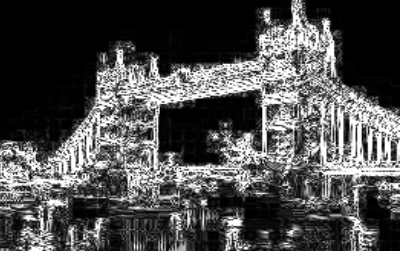
a) Orjinal Görüntü



b) Gri Tonlu Görüntü



c) Canny Kenar Bulma



d) Sobel Kenar Bulma



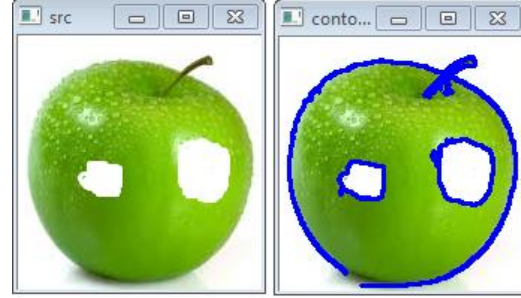
e) Laplace kenar bulma

Şekil 4. Farklı Yöntemlerle Kenar Bulma

3.3 Şekil Bulma

Herhangi bir nesnenin görüntüsünden yola çıkarak şeklini bulan Opencv fonksiyonudur. Bunun için bir

takım adımlar gerçekleştirilir. Renkli görüntü önce gri tonlu görüntüye dönüştürülür. Kenar bulma fonksiyonu ile görüntünün kenar bilgisine ulaşılır. Kenar bilgisi, şekil bulma metodunda parametre olarak kullanılarak görüntünün şekil hatları bulunur. Son olarak işaretlenen şekil hatları çizim metodu kullanılarak istenilen renkte gösterilir. (kırmızı, yeşil, mavi).

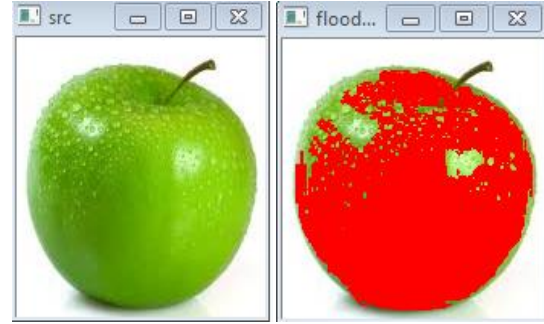


a) Kaynak Görüntü b) Görüntünün Hatları

Şekil 5. Görüntünün Şekil Hatları

3.4 Görüntü Doldurma

Herhangi bir görüntü Opencv'nin doldurma fonksiyonu ile istenilen renkte doldurulabilir. Fonksiyonda doldurma işleminin başlatılacağı piksel fonksiyona verilen parametre ile belirlenebilir. Doldurma işlemi fonksiyona verilen sınırlara göre yapılır.



Kaynak Görüntü b) Doldurulmuş Görüntü

Şekil 6. Görüntü Doldurma

3.5 Görüntü Genişletme ve Bozma

Belirli bir yapı elemanı kullanarak genişletme fonksiyonu ile bir görüntüyü genişletmek/büyütmek mümkündür. Yapı elemanı tanımlanmadığında 3x3'lük dikdörtgen biçiminde bir yapı elemanı varsayılan olarak kullanılır. Genişletme işlemi birden fazla tekrarda yapılabilir. Renkli görüntüler için her renk kanalı bağımsız olarak işlenir.

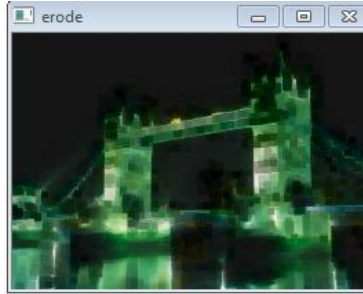
Görüntü bozma/aşındırma işlemi için ise Opencv'nin bozma fonksiyonu yine belirli veya belirtilen bir yapı elemanı ile kullanılır. Tekrar sayısı ve yapı elemanı tanımlanmadığı durumlarda görüntü genişletme işlemindeki gibi hareket edilir.



a) Kaynak Görüntü



b) Görüntü Genişletme



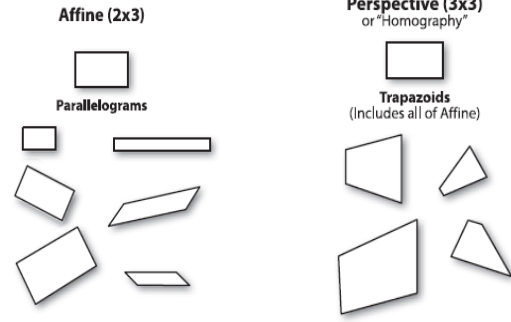
c) Görüntü Bozma

Şekil 7. Görüntü Genişletme ve Bozma

4. Geometrik Görüntü Dönüşümleri

Bir görüntüyü genişletebilen, daraltabilen, eğebilen ve döndürebilen fonksiyonlara geometrik fonksiyonlar denir. Görüntüler üzerinde geometrik dönüşümler çeşitli sebeplerle yapılabilir. Örneğin bir görüntüyü eğme veya döndürme uygulamaları sayesinde mevcut bir duvar üzerine görüntüler üst üste yerleştirilebilir. Ya da nesne tanıma uygulamaları için kullanılan görüntülerin eğitim setleri yapay olarak genişletilebilir. Geometrik dönüşümler sonucunda görüntü içeriği değişmez, ancak piksel ızgaralarının biçimleri bozulur. Düzlemsel olarak 2 çeşit geometrik dönüşüm türü vardır.

- 1- Affine Dönüşümü
- 2- Perspektif Dönüşüm



a) Affine Dönüşümü b) Perspektif Dönüşüm
Şekil 8. Affine Dönüşümü ve Perspektif Dönüşüm [1]

4.1. Affine Dönüşümü

Affine dönüşümünde 2x3'lük haritalama (eğme) matrisini temel alınmaktadır. Bu dönüşüm ile bir dikdörtgen bir paralelkenara çevrebilmektedir. Bir düzlem üzerindeki ABCD paralelkenarı Affine dönüşümü yardımıyla başka bir A'B'C'D' paralelkenarına dönüştürülebilir. Bu yapılırken kaynak görüntünün köşelerinde itme ve çekme işlemleri uygulanmaktadır. Ayrıca görüntülerin döndürülmesi ve yeniden ölçeklendirilmesi gibi özellikleri de bulunmaktadır. Bu dönüşüm sonucunda görüntünün şekli değişmekte fakat kenarları paralel kalmaktadır (Şekil 8.a).

Aynı nesnenin birkaç farklı görüntüsü karşılaştırılırken, bu farklı görünümlere sebep olan ara dönüşümleri belirlemek için affine dönüşümü kullanılabilir.

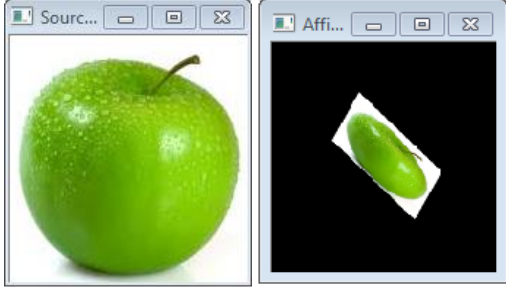
Affine dönüşümünde parametre olarak eğme ve döndürme matrisleri kullanılmaktadır. Eğme matrisinin belirlenebilmesi için kaynak ve hedef görüntünün 3 köşe noktası 3 elemanlı dizilere aktarılır. Bu 2 dizi 2 adet paralel kenar tanımlanmaktadır. Cv.GetAffineTransform() fonksiyonu ile bu diziler kullanılarak eğme matrisi elde edilir.

Cv.2DRotationMatrix() fonksiyonu ile de döndürme matrisi elde edilir. İlk parametre döndürme noktasının merkezini belirtir. Bu parametre genellikle kaynak görüntünün orta noktası olarak ayarlanır. Sonraki parametreler sırasıyla döndürme büyüklüğü ve yeniden ölçeklendirme büyüklüğüdür. Son parametre ise elde etmek istenen 2x3'lük döndürme matrisidir.

Eğer $\alpha = \text{scale} \cdot \cos(\text{angle})$ ve $\beta = \text{scale} \cdot \sin(\text{angle})$ şeklinde tanımlanırsa döndürme matrisi aşağıdaki gibi hesaplanır.

$$\begin{bmatrix} \alpha & \beta & (1-\alpha) \cdot \text{center}_x - \beta \cdot \text{center}_y \\ -\beta & \alpha & \beta \cdot \text{center}_x + (1-\alpha) \cdot \text{center}_y \end{bmatrix}$$

Şekil 9. Döndürme Matrisi



a) Kaynak Görüntü b) Affine Dönüşüm Sonucu
Şekil 10. Affine Dönüşümü

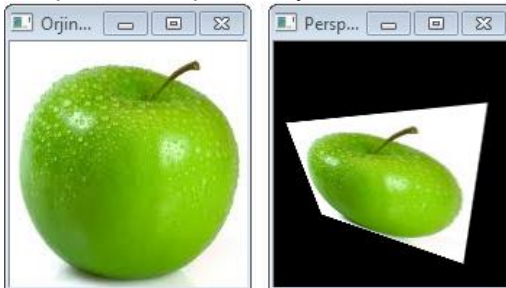
4.2 Perspektif Dönüşüm

Perspektif dönüşüm 3×3 'lük bir haritalama (eğme) matrisini temel alır. Bu dönüşüm türü ile bir dikdörtgen, eşkenar dörtgene veya yamuğa dönüştürülebilir (Şekil 8.b). Bunun dışındaki tüm özellikler affine matrisi ile benzerdir.

Eğme matrisi belirlenirken kaynak ve hedef görüntünün 4 köşe noktası 4 elemanlı dizilere aktarılır. Kaynak ve hedef görüntünün köşe noktaları bu dizilere aktarıldıktan sonra ilgili fonksiyon yardımıyla haritalama (eğme) matrisi elde edilir. Eğme matrisi elde edildikten sonra Perspektif dönüşüm işlemi gerçekleştirilir.

Perspektif dönüşümde döndürme matrisi kullanılmaz. Bu nedenle elde edilen görüntü döndürülmüş görüntü değil, eşkenar dörtgene çevrilmiş görüntüdür.

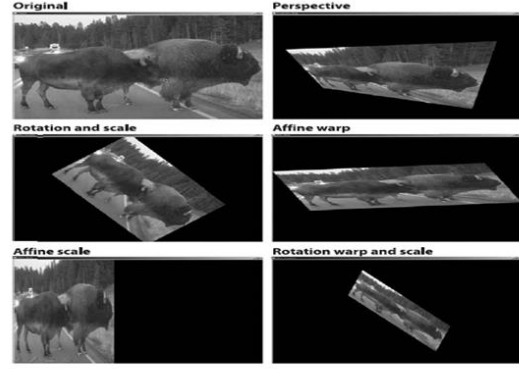
Aşağıdaki perspektif dönüşüm işlemi gerçekleştirilmiştir. Burada dikkat edilmesi gereken bir nokta da eğme matrisinin belirlenmesi için kullanılan noktaların affine dönüşümündeki gibi 3 adet değil de, 4 adet olmasıdır. Yani dikdörtgen biçimindeki kaynak ve hedef görüntünün 4 köşesi de belirtilir. Ve eğme matrisi bu noktalara göre elde edilir. Ayrıca eğme matrisi daha önce olduğu gibi 2×3 'lük değil, 3×3 'lüktür. Bu farklar dışında perspektif dönüşüm işleminin daha önce bahsettiğimiz affine dönüşümünden hiçbir farkı yoktur.



a) Kaynak Görüntü b) Perspektif Dönüşüm Sonucu

Şekil 11. Perspektif Dönüşüm

Aşağıdaki şekilde ise farklı amaçlarla elde edilen dönüşüm sonuçları gösterilmektedir.



Şekil 12. Çeşitli Affine ve Perspektif Dönüşümleri [1]

5. Kuş Bakışı Görüntü Dönüşümü

Kuş bakışı görüntü dönüşümü yatay ekseninde kaydedilen görüntüleri tepeden (yukarıdan aşağıya) çekilmiş gibi gösterebilen bir dönüşüm türüdür. Geometrik dönüşümlerden yararlanılarak gerçekleştirilir.

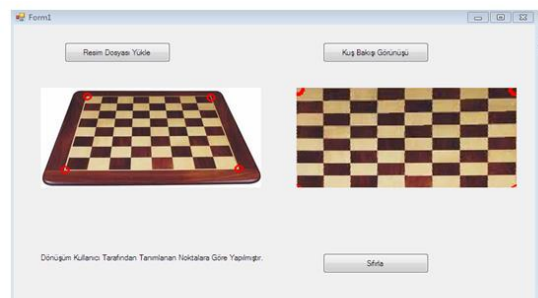
Algoritması:

- 1-) Görüntüde dönüşüm yapılacak 4 nokta belirlenir.
- 2-) Bu noktalar kullanılarak haritalama (dönüşüm) matrisi elde edilir.
- 3-) Elde edilen dönüşüm matrisi kullanılarak perspektif dönüşüm gerçekleştirilir.
- 4-) Sonuç olarak elde edilen görüntü kuş bakışı görüntüdür.

5.1 Kuş Bakışı Görüntü Dönüşüm Yazılımı

Kuş bakışı dönüşüm yazılımını gerçekleştirmek için kullanılacak fonksiyonlar OpenCV kütüphanesinde mevcuttur.

Geliştirilen program C Sharp ile yazıldığından, OpenCV ile C Sharp arasında haberleşmeyi sağlayan EmguCV kütüphanesi kullanılmıştır. Program her aşamada kullanıcıları yönlendirerek doğru sonuca gitmeyi sağlamaktadır. Programda kullanıcıdan resim dosyası yüklemesini ve ardından yüklenen resim dosyasından 4 adet nokta seçmesini istemektedir. Seçim işlemi tamamlandıktan sonra kuş bakışı dönüşümü butonuna basıldığında, belirtilen noktalar arasında kalan görüntü bölümünün dönüşüm sonucu gösterilmektedir.



Şekil 13. Kuş Bakışı Görüntü Dönüşüm Yazılımı

6. Sonuç

Bu çalışmada temel görüntü dönüşümleri incelenmiş ve OpenCV yardımıyla dönüşüm çıktıları elde edilmiştir. Geometrik dönüşümlerden bahsedilmiş, her birinin algoritmaları incelenip, OpenCV kütüphanesi ile programlanmıştır. Ayrıca farklı amaçlar doğrultusunda (döndürme, yeniden ölçeklendirme, eğme vs.) gerçekleştirilen bu dönüşümlerin sonuçları gösterilmiştir. Bu dönüşümlerden yararlanılarak, asıl amaç olan kuş bakışı görüntü dönüşümünün algoritmasını oluşturan adımlar gerçekleştirilmiştir. OpenCV kütüphanesi kullanılarak algoritma adımları geliştirilen program uygulanmış ve dönüşüm işlemi tamamlanmıştır.

7. Kaynaklar

[1] Bradski, G. and Kaehler, A., “Learning OpenCV: Computer Vision with the OpenCV Library”, O’Reilly Media, 2008, Amerika Birleşik Devletleri

[2] [Http://opencv.willowgarage.com/wiki](http://opencv.willowgarage.com/wiki)

[3] INTEL CORPORATION: Intel researchers teach computers to ‘read lips’ to improve accuracy of speech recognition software. M2 Presswire, Coventry, Apr 28,2003, pg1.

[4] Roebert, S., Schmits, T., and Visser, A., “Creating a Bird’s Eye View Map Using an Omnidirectional Camera Intelligent System” Laboratory Amsterdam Universiteit van Amsterdam, 1098 SJ Amsterdam, the Netherlands