

**Stereo Kamera Kullanılarak Görüntüde Derinlik Algılaması
ve Koordinat Hesaplaması**
Araştırma Raporu

İçerik

1 Stereo Görü	1
2 Stereo Görünün Kullanım Yöntemleri	1
3 Stereo Görü Sistemlerinin Artıları ve Eksileri	2
4 Stereo Eşleştirme (Stereo Matching)	2
5 Stereo Eşleştirme Algoritmaları	2
5.1 Özellik Tabanlı Eşleştirme	2
5.2 Bölgesel Eşleştirme	3
6 Aykırılık Haritası (Disparity Matrix)	3
7 Algoritma Akış Şeması	5
8 Stereo Görüntünün İşlenmesi ve Derinlik Kestirimi	6
9 Deneyler ve Sonuç	9
9.1 Uygulama Maliyeti ve Verimlilik Çözümlemesi	10

Şekiller Listesi

1 Stereo Görüntü Örneği	1
2 İmgecik Eşleştirme Maliyet Fonksiyonları	3
3 Bölüt Eşleştirme Maliyet Fonksiyonları	3
4 Bölge Eşleştirme	4
5 Algoritma Akış Diyagramı	5
6 Aykırılık Haritası Çıkarımı	9
7 Farklı Bölge Derinlikleri ile Aykırılık Haritası Çıkarımı	9
8 Aykırılık Haritası Çıkarımı - 2	9

Çizelgeler Listesi

1 Avantajlı ve Dezavantajlı Durumlar Tablosu	2
--	---

1 Stereo Görü

Geçmişten günümüze doğru makine görüşünün gelişimine bakıldığında araştırmacıların ilk uygulamalarında tek bir görüntü girdi kaynağı (kamera) kullandıkları görülmektedir. Zaman içerisinde bu sistemler tek kameralı oldukları için görüntü derinliği, konumlandırma (koordinat çıkartma), boyut algılama gibi birtakım istenirlere yapıları gereği cevap veremez hale gelmişlerdir. Bu nedenlerden dolayı bir bakıma insanın görme yapısı taklit edilerek birden fazla girdi görüntü kaynağı kullanılmaya başlanmıştır. Böylelikle farklı açılar ile çekilmiş görüntüler birbirleri ile kıyaslanarak görüntüler üzerinden belirli çıkarımlar yapılabilir hale gelinmiştir. Birden fazla kamera ile gerçekleştirilmiş bu sistemler stereo görüntü sistemleri olarak tanımlanmışlardır.



Şekil 1: Stereo Görüntü Örneği

2 Stereo Görünün Kullanım Yöntemleri

- Otonom Araçlar

Stereo görü sistemi aracılığı ile özdevinimli araç takip uzaklığının ayarlanması, kaza önleme sistemleri, yol tutuş sistemlerinde kullanılmaktadır.

- Topoğrafik Harita Oluşturma

Stereo görüntünün ilk kullanım alanlarından birisidir[2]. Derinlik algısının hesaplanması ile haritanın izometrik olarak modellenenmektedir.

- Robotik

Engellerin yoğun olmadığı yerlerde hareket eden robotların bir çoğunda sonar dönüştürücü veya lazer mesafe duyargaları kullanılabilmektedir. Ancak bir ev ya da kapalı mekan içerisinde aynı verimi sağlayamamaktadır. Bu durumlarda çevrenin tanımlanması, engellerden kaçınma ve mesafe algılama işlevlerinin gerçekleştirilmesinde stereo görü kullanılabilmektedir.

- Yüz Tanıma

Yüz tanıma işleminde kızılötesi taramayla yansıyan ışık fotonları ile tanımlama yapılması yerine stereo görü ile derinlik algılanması sağlanabilmektedir.

3 Stereo Görü Sistemlerinin Artıları ve Eksileri

Avantajlı Durumlar	Dezavantajlı Durumlar
1. Yaygın kullanım alanı 2. Paralel işlenmeye oldukça uygun 3. Otonom araç ve robotik uygulamaları	1. Maliyetli bir arama problemi. 2. Düşük ışık koşullarında zorluk 3. Tek kameralı sistemlere göre daha maliyetli 4. Kalibrasyon problemleri 5. Yüksek çözünürlük gereksinimi

Çizelge 1: Avantajlı ve Dezavantajlı Durumlar Tablosu

4 Stereo Eşleştirme (Stereo Matching)

Stereo eşleştirme işlemi temel anlamda bir arama problemi gibi düşünülebilir. Burada aranmak istenen şey sol görüntüdeki bir değerin ya da değerler kümesinin sağ görüntüdeki konumu ya da sağ görüntüdeki benzerinin konumudur. Böylelikle görüntüler arasındaki yer değiştirme farkı ile bir derinlik ölçüsü elde edilmeye çalışılmaktadır. Bu arama problemini zorlaştıran birkaç durum da bulunmaktadır. Örnek olarak sol görüntüde olan bir nesnenin sağ görüntüde açıdan dolayı başka bir nesnenin arkasında kalması, tek boyutlu bir görüntünün aranması (duvar, gökyüzü), kameralar arası uzaklığın gözardı edilecek kadar az olması, görüntüler üzerinde gürültü olması, görüntülerin histogram değerlerinin belirli bir noktada öbeklenmesi vs. Bu nedenlerden dolayı bazı durumlarda stereo eşitlemesi uygulamadan önce görüntülerin iyileştirilmesi de gerekebilmektedir. Bazı durumlarda gauss filtesi ve histogram eşitliği işlemlerinin stereo eşitlemeden önce uygulanması tercih edilmektedir.

5 Stereo Eşleştirme Algoritmaları

Stereo eşleştirme algoritmaları genel kapsamıyla iki ana başlık altında toplanmaktadır. Özellik tabanlı eşleştirme ve bölgesel eşleştirme algoritmaları.

5.1 Özellik Tabanlı Eşleştirme

Özellik tabanlı eşleştirme işlemleri görüntüdeki köşe, kavis, kenar vb. gibi özellikler yardımıyla eşleştirme işlemini gerçekleştirmektedir [1]. Bu tarz algoritmalar bölgesel eşleştirme algoritmalarına göre daha hızlı sonuçlar üretmektedir ancak oluşturdukları derinlik haritaları daha çok özel bir bölge için kullanılmaya uygundur.

5.2 Bölgesel Eşleştirme

Bölgesel eşleştirme tabanlı algoritmalar görüntüdeki her bir imgeciğin karşılığını diğer görüntüde arama mantığına dayanırlar. Hangi imgeciğin aranan imgeciğin karşılığı olduğunu belirlemede birkaç farklı maliyet fonksiyonu bulunmaktadır. Bunlardan bazıları şunlardır:

- Mutlak Fark (Absolute Differences - AD) : Referans imgeciyle bakılan imgeciğin farkının mutlak değeri.
- Karesi Alınmış Fark (Sum of Squared Differences - SSD): Referans imgeciyle bakılan imgeciğin farkının karesi.

$$C_{AD}(x, y, disparity) = |I_{left}(x, y) - I_{right}(x, y + disparity)|$$
$$C_{SSD}(x, y, disparity) = |I_{left}(x, y) - I_{right}(x, y + disparity)|^2$$

C = Maliyet Fonksiyonu
 I_{left} = Sol Görüntü
 I_{right} = Sağ Görüntü

Şekil 2: İmgecik Eşleştirme Maliyet Fonksiyonları

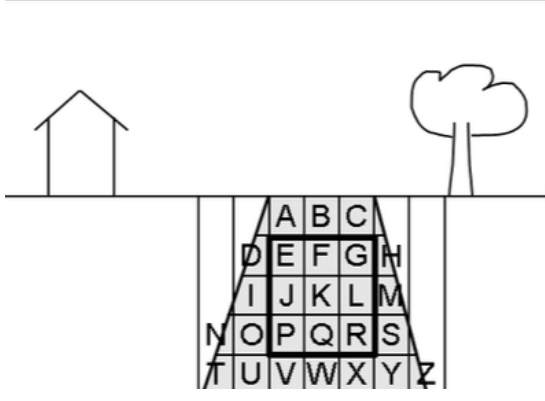
Her bir sol görüntüdeki imgecik için maliyet fonksiyonun çıktısına göre sol görüntüdeki imgeciğin sağ görüntüde ne kadar sapma ile bulunduğu hesaplanır ve bu sapma değerleri ile bir sapma matrisi (disparity matrix) oluşturulur. Algoritmanın performansını iyileştirmek için genellikle imgeciğin yanındaki komşu imgeciklerde arama işlemine eklenir. Bu işlem için bir pencere boyutu belirlenerek (örn. 3x3, 7x7) pencerenin merkezine aykırılık değeri hesaplanacak olan imgecik yerleştirilir. Şekil 3’de görüldüğü gibi daha sonrasında pencerenin içinde kalan tüm imgecikler ile bir maliyet çıkarımı yapılır.

```
foreach element in block do  
  | Calculate  $C_{AD}(x, y, disparity)$  as  $|I_{left}(x, y) - I_{right}(x, y + disparity)|$ ;  
end  
foreach element in block do  
  | Calculate  $C_{SSD}(x, y, disparity)$  as  $|I_{left}(x, y) - I_{right}(x, y + disparity)|^2$ ;  
end
```

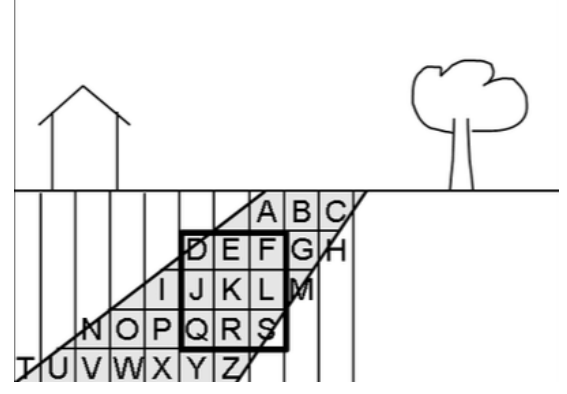
Şekil 3: Bölüt Eşleştirme Maliyet Fonksiyonları

6 Aykırılık Haritası (Disparity Matrix)

Aykırılık haritası sol taraftaki görüntünün sağ tarafta ne kadar sapma ile karşılığının bulunabildiğini anlatan bir matristir. Örnek olarak $[i, j]$ noktasını merkez alan bir sol görüntü bölgesinin karşılığı sağ görüntüde $[i, j + p]$ noktasını merkez alan bir bölgede bulunuyorsa aykırılık



(a) Sol Görüntü

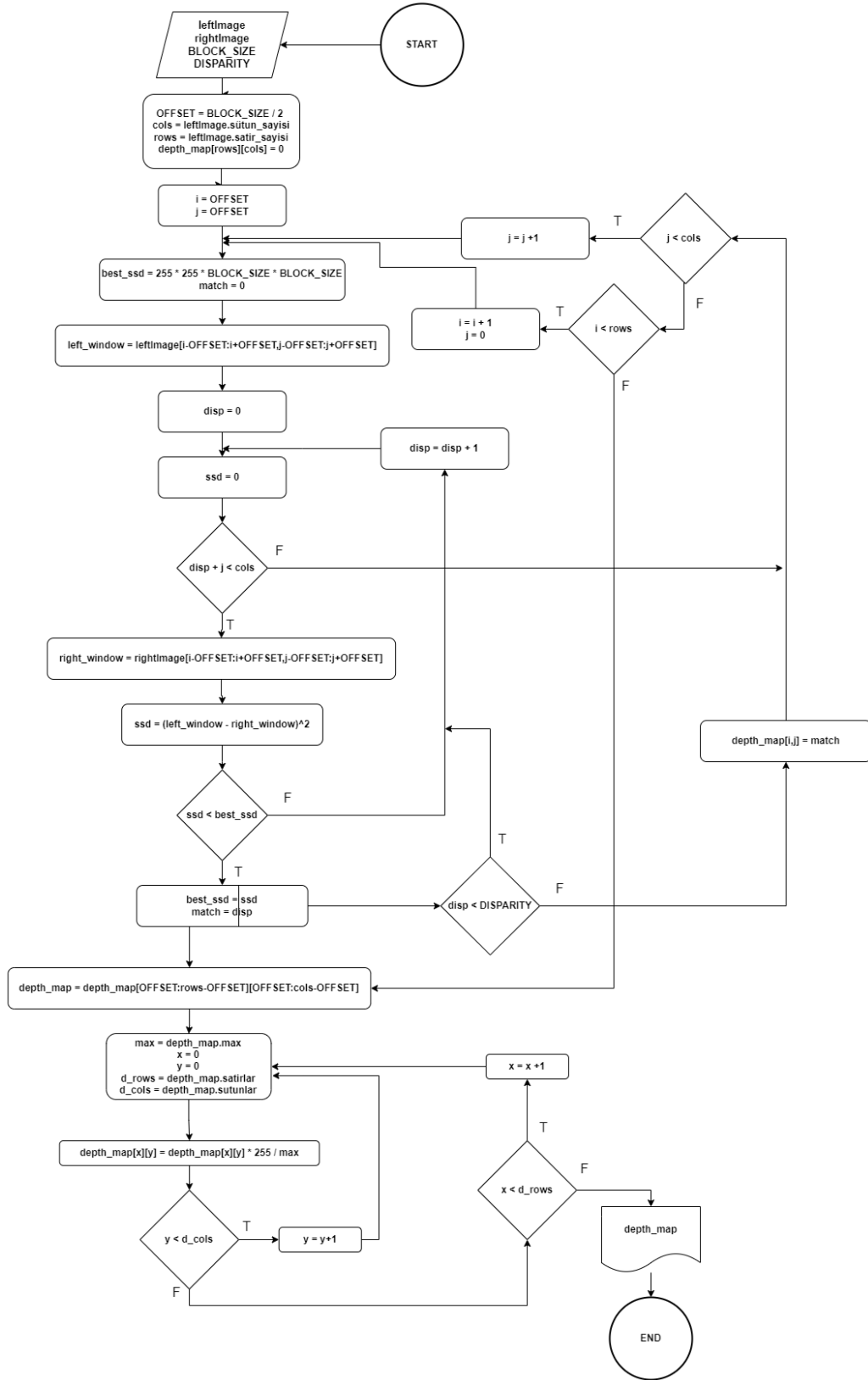


(b) Sağ Görüntü

Şekil 4: Bölge Eşleştirme

matrisinin $[i][j]$ değeri p 'ye eşit olacaktır. Eğer görüntünün bir noktadaki sapma değeri yüksekse o noktanın kameraya daha yakın olduğu anlaşılır. Bunun nedeni cisimler uzaklaştıkça izdüşümleri küçüldüğü için kameraların konumundan dolayı olan sapma miktarı da azalacaktır. Basit bir deyişle kameraya yakın olan cisimler iki görüntü arasında daha fazla sapacaktır. Aykırılık haritasında yakındaki cisimler beyaz tonlara daha yakın olacaktır.

7 Algoritma Akış Şeması



Şekil 5: Algoritma Akış Diyagramı

8 Stereo Görüntünün İşlenmesi ve Derinlik Kestirimi

```
#include <opencv2/opencv.hpp>

using namespace cv;
using namespace std;

#define BLOCK_SIZE 15
#define OFFSET (BLOCK_SIZE/2)
#define DISPARITY_RANGE 64

unsigned char left_window[BLOCK_SIZE][BLOCK_SIZE] = { 0 };
unsigned char right_window[BLOCK_SIZE][BLOCK_SIZE] = { 0 };

Mat leftImage = imread("C:\\Users\\ben\\Desktop\\hm2.png",
    IMREAD_GRAYSCALE);
Mat rightImage = imread("C:\\Users\\ben\\Desktop\\hm1.png",
    IMREAD_GRAYSCALE);

int calculate_ssd_of_blocks() {
    int diff = 0;
    int ssd = 0;
    for (int i = 0; i < BLOCK_SIZE; i++) {
        for (int j = 0; j < BLOCK_SIZE; j++) {
            diff = left_window[i][j] - right_window[i][j];
            ssd += diff * diff;
        }
    }
    return ssd;
}

void load_left_window(int row, int col, Mat img) {
    for (int x = 0; x < BLOCK_SIZE; x++) {
        for (int y = 0; y < BLOCK_SIZE; y++) {
            left_window[x][y] = img.at<uchar>(row + x - OFFSET, col
                + y - OFFSET);
        }
    }
}

void load_right_window(int row, int col, Mat img) {
    for (int x = 0; x < BLOCK_SIZE; x++) {
```



```

        for (int y = 0; y < BLOCK_SIZE; y++) {
            right_window[x][y] = img.at<uchar>(row + x - OFFSET, col
                + y - OFFSET);
        }
    }
}

int main() {
    Mat depthMap = leftImage.clone();
    int cols = leftImage.cols;
    int rows = leftImage.rows;

    for (int i = OFFSET; i < rows - OFFSET; i++) { //search all the
        pixels except borders
        for (int j = OFFSET; j < cols - OFFSET; j++) {
            int best_ssd = 255 * 255 * BLOCK_SIZE * BLOCK_SIZE;
            int match = 0;

            load_left_window(i, j, leftImage);

            for (int disp = 0; disp < DISPARITY_RANGE; disp++) {
                //compare left block with right block
                int ssd = 0;
                if (j + disp < cols - OFFSET) {
                    load_right_window(i, j + disp, rightImage);
                    ssd = calculate_ssd_of_blocks();
                    if (ssd < best_ssd) {
                        best_ssd = ssd;
                        match = disp;
                    }
                }
            }
            depthMap.at<uchar>(i,j) = static_cast<uchar>(match);
        }
    }

    Mat croppedDepthMap(depthMap(Range(OFFSET, rows - OFFSET),
        Range(OFFSET, cols - OFFSET)));
    Mat normalizedDepthMap;

```

```
normalize(croppedDepthMap, normalizedDepthMap, 255, 0,
         cv::NORM_MINMAX);
normalizedDepthMap.convertTo(normalizedDepthMap, CV_8UC1);
imshow("Normalize Derinlik Haritasi", normalizedDepthMap);
imshow("Derinlik Haritasi", croppedDepthMap);
imwrite("test.jpg", normalizedDepthMap);
waitKey(0);

return 0;
}
```

9 Deneyler ve Sonuç

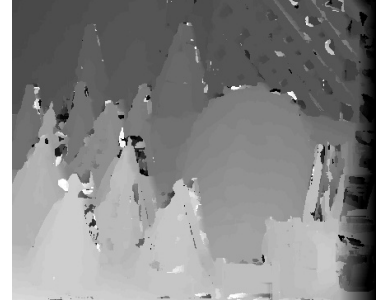
İki farklı görüntü üzerinde bölge eşitleme algoritmaları denenmiştir. Maliyet fonksiyonu hesabında SSD yöntemiyle aykırılık haritası oluşturulmuştur. Arama aralığı 64 birim seçilmiştir. Çıktı görüntüleri olarak aykırılık haritası 0 ile 255 arasında normalize edilmiştir.



(a) Sol Görüntü

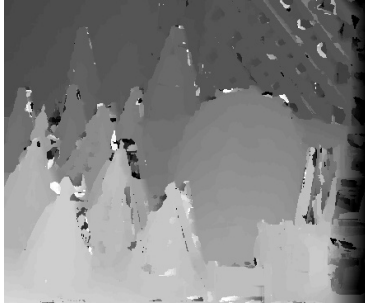


(b) Sağ Görüntü

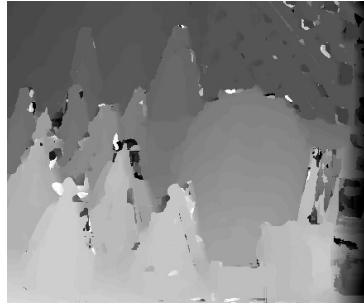


(c) Bölge Derinliği = 9

Şekil 6: Aykırılık Haritası Çıkarımı



(a) Bölge Derinliği = 9



(b) Bölge Derinliği = 11



(c) Bölge Derinliği = 15

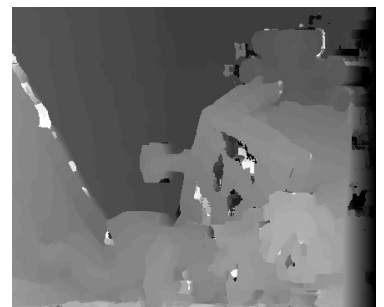
Şekil 7: Farklı Bölge Derinlikleri ile Aykırılık Haritası Çıkarımı



(a) Sol Görüntü



(b) Sağ Görüntü



(c) Bölge Derinliği = 15

Şekil 8: Aykırılık Haritası Çıkarımı - 2

Geliştirilen algoritma temel openCV işlevine göre daha verimsiz çıktılar üretmektedir. Beklenildiği üzere bölge derinliğinin artması durumunda algoritmanın çıktısı daha da düzenlileşmektedir. Çıktılar üzerinde derinlik algısının yakalanabileceği renk ayrımları elde edilebilmiştir ancak bazı durumlarda derinlik ayrımının daha net olması beklenen durumlarda beklenen sonuçlar elde edilememiştir.

9.1 Uygulama Maliyeti ve Verimlilik Çözümlemesi

Uygulama geliştirilirken araştırmanın verimli ve kuramsal olarak araştırmacıya katkı sağlaması için olabildiğince en temel işlemler hazır kütüphane işlevleri kullanılmadan gerçekleştirilmiştir. Uygulama, hazır fonksiyonlar ile kıyaslandığında büyük bir hız farkı gözlemlenmiştir. Bunun başlıca nedeni bölge eşitleme algoritmalarının günümüz işlemcilerinin tek çekirdek-tek iş parçacığı desteğiyle uygun bir sürede kaldıramayacağı kadar ağır olmasıdır. Bölge eşitleme işlemleri yüksek aritmetik desteğe ihtiyaç duyduğundan dolayı sınırlı kaynağa sahip olan sistemlerde (IHA, otonom araçlar vs.) kullanılmak istenildiğinde daha yüksek batarya tüketim miktarlarına veya darboğaz sorunlarına yol açabilirler. Bu uygulamanın hareketli bir cisim üzerinde stereo kamera ile gerçekleştirilmesi büyük olasılıkla düşük kare hızı çıktısı verecektir. Bu eksikliğin yardımcı bir aritmetik birim desteği ile iyileştirilmeye çalışılması bu seferde batarya sorunlarına yol açacaktır. Birden fazla kamera kullanım durumu da beraberinde kalibrasyon ve birim dönüşüm sorunları da doğuracaktır.

References

- [1] M. O. Özcan. Stereo video görüntülerini işlemeye dayalı derinlik analizi. Master's thesis, Trakya Üniversitesi Fen Bilimleri Enstitüsü, 2014.
- [2] R. Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.