

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

OPTİK AKIŞ İLE HAREKET TESPİTİ

Elektronik ve Haberleşme Mühendisi Eda ÖZÜNTÜRK

**FBE Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Haberleşme Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Prof. Dr. Vedat TAVŞANOĞLU

İSTANBUL, 2007

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	iv
KISALTMA LİSTESİ.....	v
ŞEKİL LİSTESİ.....	vi
ÖNSÖZ.....	ix
ÖZET.....	x
ABSTRACT.....	xi
1. GİRİŞ	1
1.1 İnsan Görsel Sistemi	1
1.2 Nesne Tanıma	3
1.3 Hareket Alanı Tanımı	4
1.4 Optik Akış Tanımı.....	7
1.4.1 İki Boyutlu Hareket Kısıtlama Denklemi.....	9
1.4.2 Belirsizlik Problemi (Aperture Problem)	15
1.5 Optik Akış Teknikleri.....	17
2. FARK TEKNİKLERİ.....	18
2.1 Fark Tekniklerinin Genel Yapısı.....	19
2.1.1 Fark Alma.....	19
2.1.2 Akış Hızı Laplasiyenleri Hesabı.....	20
2.2 Horn ve Schunck Optik Akış Modeli.....	21
2.2.1 Yumuşatma Bileşeni.....	21
2.2.2 Hataları Minimize Etme.....	21
2.2.3 Tekrarlama Yöntemi.....	22
2.2.4 Horn ve Schunck Algoritması ile MATLAB Uygulaması.....	25
2.2.4.1 Test Sonuçları.....	26
2.3 Lucas ve Kanade Optik Akış Modeli.....	31
2.3.1 Kaydetme Problemi.....	31

2.3.2	Kaydetme Algoritması.....	32
2.3.2.1	Tek Boyutlu Durum.....	32
2.3.2.2	Alternatif Yaklaşım.....	35
2.3.2.3	Performans.....	36
2.3.2.4	Uygulama.....	37
2.3.2.5	Çok Boyutlu Gerçekleme.....	38
2.3.3	Minimize Etme.....	39
2.3.4	Temel Lucas Kanade Metodu MATLAB Programı Uygulaması.....	42
2.3.4.1	Test Sonuçları.....	42
2.3.5	Lucas Kanade Yönteminin Gauss Piramidi Uygulaması.....	47
2.3.5.1	Gauss Ölçek-Uzayı (Piramidi)	47
2.3.5.2	Ölçek-Uzayı Tanımı.....	47
2.3.5.3	Problem Tanımı.....	52
2.3.5.4	Hız Tahmini Algoritması.....	53
2.3.5.5	Görüntü Piramidi Gösterimi.....	53
2.3.5.6	Piramit Hız Tahmini.....	55
2.3.5.7	Tekrarlamalı Lucas ve Kanade Optik Akış Hesaplaması.....	57
2.3.5.8	Lucas Kanade Yönteminin Gauss Piramidi Algoritması.....	61
2.3.5.9	Test Sonuçları.....	63
2.3.6	Lucas Kanade Uygulamalarında Karşılaşılan Hatalar.....	68
3.	SONUÇLAR ve ÖNERİLER.....	69
	KAYNAKLAR	70
	ÖZGEÇMİŞ.....	71

SİMGE LİSTESİ

I_x, I_y	Uzaysal kısmi türevler
I_t	Zamansal kısmi türevler
ij	Grid noktası belirteci
v_x, v_y	x ve y yönündeki hız bileşenleri
$\bar{v}_n$	Normal hız bileşeni
$\hat{n}$	Birim normal hız doğrultusu
v_n	Birim normal hız genliği
Δ	Laplasiyen operatörü
∇	Gradyan operatörü
I	Görüntü parlaklığı
α	Yumuşatma terimi
E_b	Görüntü parlaklığı değişimdeki hatalar
E_c	Hız akışındaki sapma hataları
$\bar{v}_x, \bar{v}_y$	x ve y yönündeki ortalama hızlar
E	Toplam hata
k	Tekrarlama sayısı
v_x^0, v_y^0	x ve y yönündeki başlangıç hızları
$\vec{v}$	Hız vektörü
W	Pencere fonksiyonu
Ω	Görüntü üzerinde ilgilenilen alan

KISALTMA LİSTESİ

HS	Horn ve Schunck
LK	Lucas ve Kanade
OKH	Ortalama Küçük Hatalar
YDT	Yüksek Dereceden Terimler

ŞEKİL LİSTESİ

Şekil 1.1 İnsan gözü yapısı.....	1
Şekil 1.2 Elektromanyetik spektrum.....	3
Şekil 1.3 Pinhole kamera modeli; bazı 3 boyutlu noktalar ve bu noktaların karşılık gelen görüntü noktaları.....	6
Şekil 1.4 (a) İleri hareket, (b) dönme, (c) yatay çevrim, (d) yakın objeler daha hızlı hareket ediyor görünür.....	7
Şekil 1.5 (a) Bir iniş alanında belirli bir noktaya inerken ileriye bakan bir pilotun hareket alanı. (b) Pilotun sağa bakarken gördüğü hareket alanı.....	7
Şekil 1.6 Direğin optik akışı ve hareket alanı.....	8
Şekil 1.7 Belirsizlik problemi. Sadece b bileşeni hesaplanabilir.....	8
Şekil 1.8 (x,y,t) konumundan, $(x+dx, y+dy, t+dt)$ konumuna hareket eden noktanın şiddet değişmezliği.....	10
Şekil 1.9 Görüntüde bir noktanın yer değiştirmesinin zamana bağlı olarak hız bileşenleri ile gösterilmesi.....	10
Şekil 1.10 (a) Belirsizlik problemi, (b) Hareket Kısıtlama denkleminin normal gelen hız.....	13
Şekil 1.11 İki Boyutlu Hareket Kısıtlama Denkleminin grafiksel gösterimi.....	14
Şekil 1.12 (a) t anında küçük bir mesafe boyunca bir doğru gözlenmektedir. (b) $(t+dt)$ anında, doğrunun her noktasının nereye hareket ettiğine tam olarak karar verilmesi.....	15
Şekil 1.13 (a) Sabit ışık kaynağına sahip dönen küre (b) Hareket eden ışık kaynağına sahip sabit küre.....	16
Şekil 2.1 Türev hesaplamalarında kullanılan sekiz ardışık ölçüme sahip küp.....	19
Şekil 2.2 Laplasiyen hesabında kullanılan ağırlıklar.....	20
Şekil 2.3 Kısıtlama doğrusuna dik olarak akış hızının lokal ortalamasından çizilen doğru bulunan, hatayı indirgeyen akış hızı değeri	24
Şekil 2.4 Horn ve Schunck yöntemi ile optik akış hesabı blok şeması.....	26

Şekil 2.5 (a), (b) Orijinal görüntü dizileri.....	27
Şekil 2.6 (a) x yönündeki kısmi türev, (b) y yönündeki kısmi türev, (c) t yönündeki kısmi türev.....	28-29
Şekil 2.7 (a) Optik akış alanı, (b) enerji alanı.....	30
Şekil 2.8 (a), (b) Görüntü kaydetme problemi.....	31-32
Şekil 2.9 Eşleştirilen iki eğri.....	33
Şekil 2.10 Eşleştirilen eğrilerin h mesafesine göre değişimleri.....	35
Şekil 2.11 Lucas-Kanade metodunun blok diyagramı.....	42
Şekil 2.12 (a), (b) Orijinal görüntü dizileri.....	43
Şekil 2.13 5x5 pencere için (a) Optik akış alanı, (b) Enerji alanı.....	44
Şekil 2.14 10x10 pencere için (a) Optik akış alanı, (b) Enerji alanı.....	45
Şekil 2.15 15x15 pencere için (a) Optik akış alanı, (b) Enerji alanı.....	46
Şekil 2.16 (a) Ölçek-uzayı seviyesi $L(x,y,0)$, f görüntüsüne karşılık gelir.....	48
Şekil 2.16 (b) Ölçek-uzayı seviyesi $L(x,y,1)$	48
Şekil 2.16 (c) Ölçek-uzayı seviyesi $L(x,y,4)$	49
Şekil 2.17 Aşağı örnekleme gösterimi.....	49
Şekil 2.18 Örnek görüntünün 4 seviyeli ($L(x,y,4)$) Gauss Piramidi uygulaması.....	50
Şekil 2.19 Örnek görüntünün Gauss Piramidi uygulamasının 2 seviyesi için aynı boyuttaki gösterimi.....	51
Şekil 2.20 512x512'lik görüntünün 9 seviyeli ($L(x,y,9)$) Gauss Piramidi gösterimi.....	52
Şekil 2.21 Görüntünün t ve t+1 anındaki iki çerçevesi için Gauss Piramidi gösterimi.....	63
Şekil 2.22 Orijinal görüntü dizileri (Hamburg Taksi)	64
Şekil 2.23 (a) En üst piramit seviyesinde yer alan temel görüntü dizileri.....	64
Şekil 2.23 (b) İkinci piramit seviyesinde yer alan görüntü dizileri.....	64
Şekil 2.23 (c) En alt piramit seviyesinde yer alan görüntü dizileri.....	65

Şekil 2.24 (a) En üst piramit seviyesinden elde edilen hız vektörleri.....	66
Şekil 2.24 (b) İkinci piramit seviyesinden elde edilen hız vektörleri.....	66
Şekil 2.24 (c) En alt piramit seviyesinden elde edilen hız vektörleri.....	67

ÖNSÖZ

Tez çalışmamda benden yardımlarını hiçbir zaman esirgemeyen değerli öğretmenim sayın Prof. Dr. Vedat Tavşanoğlu'na, verdikleri destekten ve yardımlarından dolayı değerli araştırma görevlisi arkadaşlarım Evren Cesur ve Nerhun Yıldız'a, sevgili arkadaşlarım Caner Arda, Ferdi Tekçe, Nazan Gönen ve Umut Gürkan'a ve her zaman sabırla yanımda olan aileme teşekkür ederim.

ÖZET

Bilgisayarla görmenin önemli alanlarından biri olan hareket tespiti; güvenlik sistemleri, trafik gözetimi gibi günlük hayatta bir çok uygulaması bulunmaktadır.

Hareket tespitinde kullanılan ilk yöntemlerden olan “Optik Akış”, hala günümüzde ortaya atılan bir çok yöntemden daha sağlam ve doğru sonuçlar üretmektedir. Ayrıca, bir çok hareket tespiti yöntemi “Optik Akış” temeline dayanmaktadır.

Tezin ilk bölümünde optik akışa genel bir bakış yapılarak temel kavramlardan bahsedilmiştir. Daha sonra optik akış yöntemlerinden ve genel yapısından bahsedilmiştir. Ayrıca optik akışın doğru olarak algılanmadığı özel durumlardan söz edilmiştir.

Hareket tespitinde kullanılan “Optik Akış” yöntemleri üç grup altında toplanmaktadır: 1. Fark teknikleri, 2. Frekans temelli teknikler ve 3. Eşleştirme teknikleri. Tezin ikinci bölümünde bu gruplardan biri olan Fark teknikleri üzerinde durulmuştur: genel fark yaklaşımı (Horn ve Schunck yöntemi) ve yerel fark yaklaşımı (Lucas ve Kanade yöntemi). MATLAB uygulamaları ile gerçek görüntü dizilerinden elde edilen hareket tahminleri vektörel ve enerji yoğunlukları çıktıları ile gösterilmiştir. Ek olarak Gauss Piramidi yönteminden bahsedilmiş, ve Lucas ve Kanade yöntemi Gauss Piramidi kullanılarak daha sağlam ve doğru sonuçlar elde edilmiştir.

Anahtar Kelimeler: Optik akış, fark teknikleri, hareket tespiti.

ABSTRACT

Motion detection which is one of the important fields in computer vision has many applications in daily life such as security systems, traffic surveillance.

“Optical Flow”, one of the first proposed approaches used for motion detection, still gives more robust and accurate results than many approaches proposed lately. Moreover, many motion detection methods are based on “Optical Flow”.

The first part of this thesis introduces fundamental concepts within general survey. Then optical flow methods and their structures are introduced. Furthermore, special cases that “Optical Flow” can not be determined accurately are mentioned.

Optical flow techniques can be classified into three groups: 1. Differential techniques, 2. Frequency-based techniques and 3. Matching techniques. Second part of this thesis introduces Differential techniques: global differential approach (Horn and Schunk method) and local differential approach (Lucas and Kanade method). Motion estimations which are obtained from MATLAB implementations using real image sequences are shown in vector fields and energy fields. Additionally, Gaussian Pyramid is discussed and with using Gaussian Pyramidal implementation of Lucas and Kanade method, more robust and accurate results are obtained.

Keywords: Optical flow, differential techniques, motion detection.

1. GİRİŞ

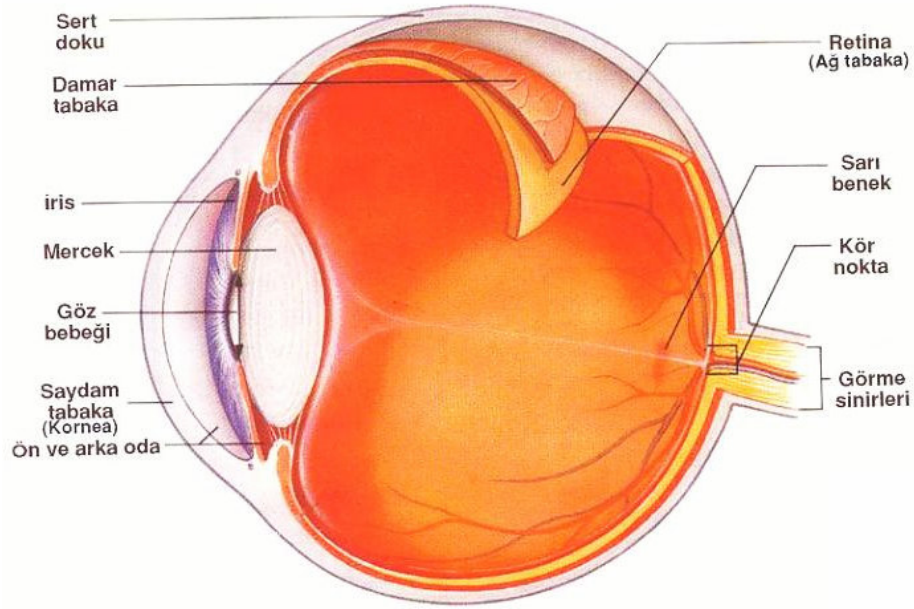
Bu tezin amacı, günümüzde kullanılan bir çok hareket tespiti yönteminin temelini oluşturan “Optik Akış” fark tekniklerini incelemektir.

Hareket tespiti yöntemlerinden bahsetmeden önce, ilk olarak nesnenin gerçekte ne olduğu tanımlanmalıdır. Nesnenin doğasını anlamak için, insan gözünün yapısı ve nesneyi biçimlendirmek için beyin ile nasıl koordineli çalıştığı üzerinde durulmalıdır. Bu nedenle ilk bölümde biyolojik sistemlerin nesneleri nasıl algıladığından bahsedilmiştir. Daha sonra optik akış ve hız alanı kavramları ele alınmış, aralarındaki ilişkinin belirlenmesi için kullanılan yöntemler üzerinde durulmuştur.

İkinci bölümde ise görüntü dizilerinden yararlanılarak hareket tahmininde karşılaşılan problemler ve bu problemlerin çözümü için öngörülen “Optik Akış” yöntemleri incelenmiştir.

1.1 İnsan Görsel Sistemi

İnsan görsel sisteminin nasıl çalıştığı teorisinin tanıtılması için, göz ve beyin arasındaki bağlantı tümüyle anlatılmalıdır (Marcus Nyström, 2003).

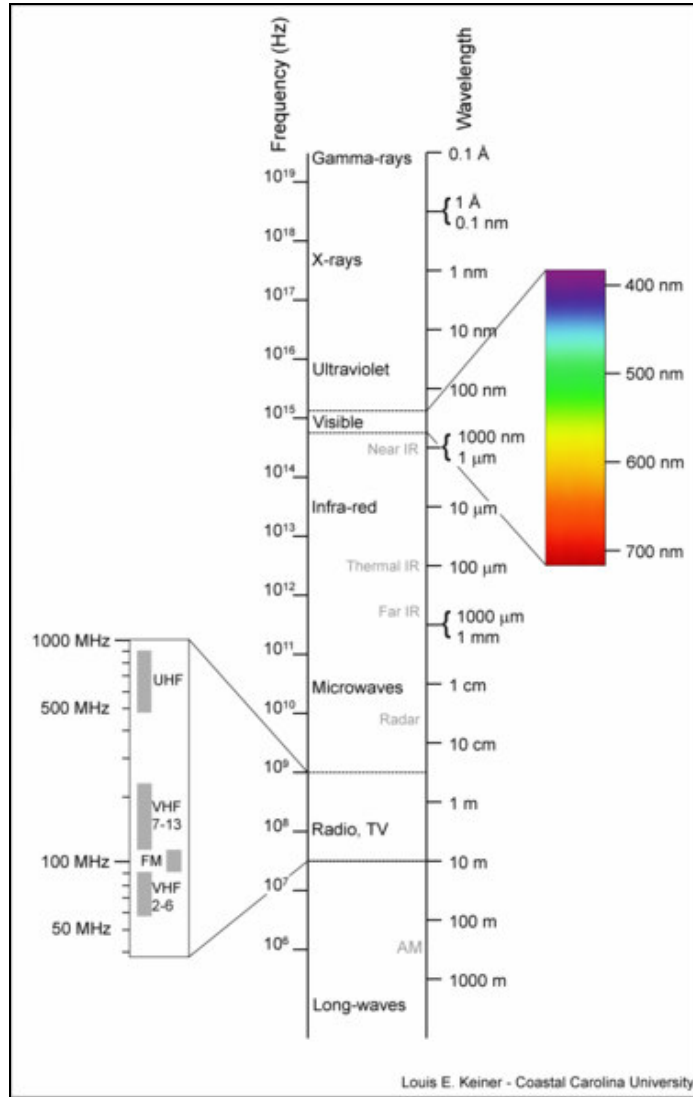


Şekil 1.1 İnsan gözü yapısı.

Saydam tabakaya giren ışık, lens tarafından retina üzerine odaklanır. Retina iki tip reseptörden oluşmaktadır: çomak ve koni. Çomak; ışıktaki küçük değişimlere çok duyarlıdır

fakat renkteki deęişimlere karşı duyarsızdır. Koni, çomağın tam tersine ışıktaki deęişimlere duyarsız fakat renk deęişimlerine duyarlıdır. Reseptörler ışığı algıladıklarında, bir sinir dürtüsü üreten kimyasal bir reaksiyon meydana gelir. Üretilen bu sinir dürtüsü, görme sinirleri demeti üzerinden bilginin yorumlandığı beyine gönderilir. Sinir yolları; sol görsel alanda yer alan bilginin sağ beyin küreciğı tarafından işlenmesini ve sağ görsel alanda yer alan bilginin sol beyin küreciğı tarafından işlenmesini sağlayacak şekilde düzenlenmiştir. Sinir dürtülerinin nasıl ilerlediğı modern bilim tarafından hala tamamıyla belgelenememiştir. Genellikle bir çok yaklaşım, modelleme amaçları için gözün mekanik özelliklerini temel almaktadır.

Nesnelerin görülebilmesi için tanımlanması gereken en basit koşul; nesnelerin ışık yaymalarıdır. Işık, nesnenin kendisinden veya nesneden yansıma ile gelebilir. Burada bahsedilen ışık, elektromanyetik (EM) spektrumun çok küçük bir bölümünü içermektedir; göz sadece bu bölümdeki deęişimlere cevap vermektedir (bakınız Şekil 1.2). Ayrıca insan gözü, farklı dalga boylarındaki ışığı algılama yetisine sahiptir; bu şekilde renkler algılanmaktadır.



Şekil 1.2 Elektromanyetik spektrum.

1.2 Nesne Algılama

İnsan beyni, farklı nesneleri algılayabilmek için bir çok problemle karşılaşmaktadır. En büyük problemlerden biri; 3 boyutlu bilginin 2 boyutlu retina üzerine iz düşürülmesidir. Bu problem nedeniyle, 2 boyutlu bilgiye bağlı olarak doğru görsel izlenimin elde edilmesi için beyin daha sonra işlemi tersine çevirmektedir.

Beyin, 3 boyutlu bilgiyi algılamak için doku gradyeni bilgisini kullanmaktadır.. Derinliği algılamakta yardımcı başka bir faktör ise “stereopsis”dir; bu faktör, gözlerin nesneleri farklı açılardan görmesi gerçeğine dayanmaktadır. Uzak mesafedeki nesnelerin retina üzerinde daha

yavaş hareket etmesi de (bu durum genellikle hareket paralaksı olarak adlandırılır) dikkate alınması gereken başka bir önemli faktördür.

Son yıllarda gelişen teoriler, bilginin algılanması ve yorumlanmasında genellikle nesne bölütlemesinin de gerekli olduğunu vurgulamaktadır.

1.3 Hareket Alanı Tanımı

Bilgisayarla görmede hareket alanı, 3 boyutlu hareketin bir kamera görüntüsüne iz düşürülmesi olarak tanımlanır. Basit bir kamera modelinde, görüntü üzerindeki her nokta (y_1 , y_2), 3 boyutlu sahnedeki bazı noktaların izdüşümüdür fakat uzaydaki sabit bir noktanın iz düşümünün yeri zamanla değişebilir. Hareket alanı, görüntü koordinatlarını 2 boyutlu bir vektöre eşleştiren bir fonksiyon olarak gösterilebilir.

Hareket alanı, iz düşümü alınan 3 boyutlu hareketin ideal tanımıdır; bu alan teorik olarak tanımlanabilir fakat pratikte görüntü verisinden sadece bir hareket alanı yaklaşımı elde edilebilir.

Basit bir kamera modeli tanımı ile hareket alanını açılabiliriz: bazı adresleme fonksiyonlarına m_1 , m_2 göre; 3 boyutlu uzayda yer alan her noktayı (x_1 , x_2 , x_3), 2 boyutlu bir görüntü noktasına (y_1 , y_2) iz düşümü alınmaktadır.

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} m_1(x_1, x_2, x_3) \\ m_2(x_1, x_2, x_3) \end{pmatrix} \quad (1.1)$$

Kamera tarafından resmedilen sahnenin zamana göre değiştiği varsayımı yapılmaktadır; sahne, birbirine göre göreceli olarak hareket eden nesnelerden oluşmaktadır.

Zamana göre bir önceki ifadenin farkını alma aşağıdaki sonucu vermektedir:

$$\begin{pmatrix} \frac{dy_1}{dt} \\ \frac{dy_2}{dt} \end{pmatrix} = \begin{pmatrix} \frac{dm_1(x_1, x_2, x_3)}{dt} \\ \frac{dm_2(x_1, x_2, x_3)}{dt} \end{pmatrix} = \begin{pmatrix} \frac{dm_1}{dx_1} & \frac{dm_1}{dx_2} & \frac{dm_1}{dx_3} \\ \frac{dm_2}{dx_1} & \frac{dm_2}{dx_2} & \frac{dm_2}{dx_3} \end{pmatrix} \begin{pmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{pmatrix} \quad (1.2)$$

Bu denklemde,

$$\vec{u} = \begin{pmatrix} \frac{dy_1}{dt} \\ \frac{dy_2}{dt} \end{pmatrix} \quad (1.3)$$

ifadesi hareket alanıdır ve $\vec{u}$ vektörünün hem görüntü konumuna, (y_1, y_2) hem de zamana, t bağlı olduğuna dikkat edilmelidir.

Benzer şekilde,

$$x' = \begin{pmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{pmatrix} \quad (1.4)$$

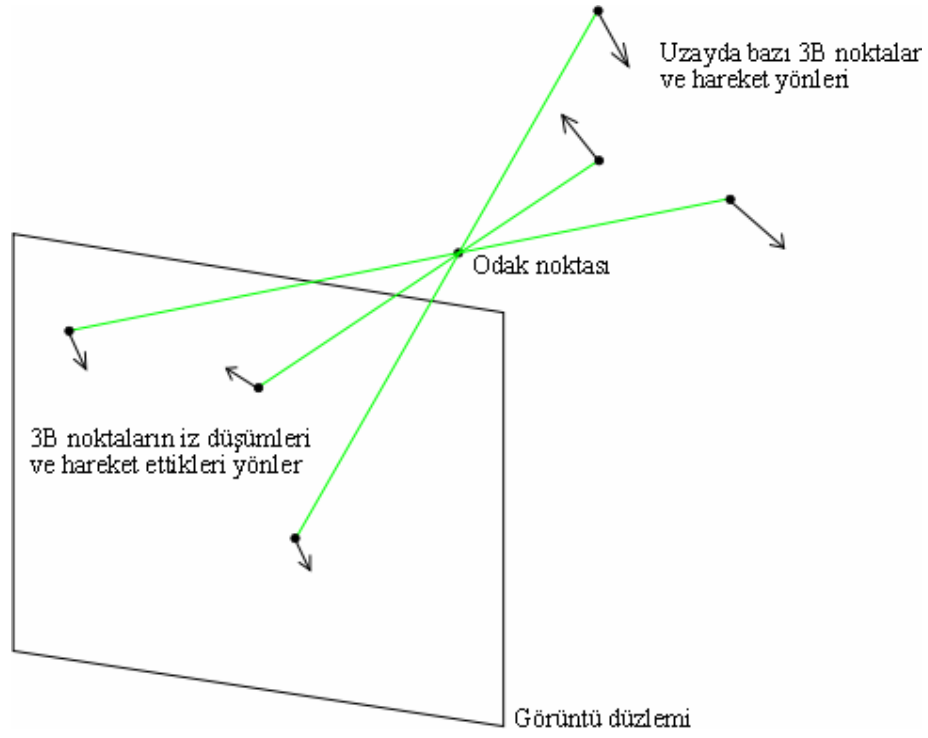
ifadesi ilişkin 3 boyutlu noktanın hareketidir ve hareket alanı ile ilişkisi şu şekilde gösterilmektedir:

$$u = Mx' \quad (1.5)$$

burada M , görüntü konumuna bağımlı 2×3 'lük matristir:

$$M = \begin{pmatrix} \frac{dm_1}{dx_1} & \frac{dm_1}{dx_2} & \frac{dm_1}{dx_3} \\ \frac{dm_2}{dx_1} & \frac{dm_2}{dx_2} & \frac{dm_2}{dx_3} \end{pmatrix} \quad (1.6)$$

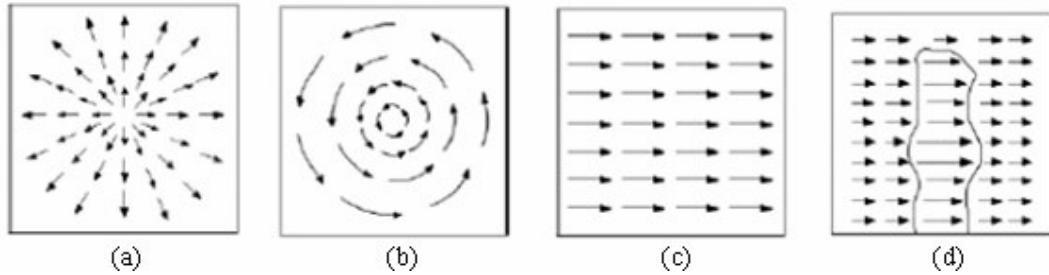
bu ilişki, belirli bir görüntü noktasında, hareket alanının M' nin sıfır uzayında yer alan 3 boyutlu hareketlere göre değişmez olduğunu gösterir. Pinhole kamera örneğinde, kameranın odak noktasından çıkan veya odak noktasına doğru yönlendirilmiş tüm 3 boyutlu hareket bileşenleri hareket alanı içerisinde tespit edilemez.



Şekil 1.3 Pinhole kamera modeli; bazı 3 boyutlu noktalar ve bu noktaların karşılık gelen görüntü noktaları.

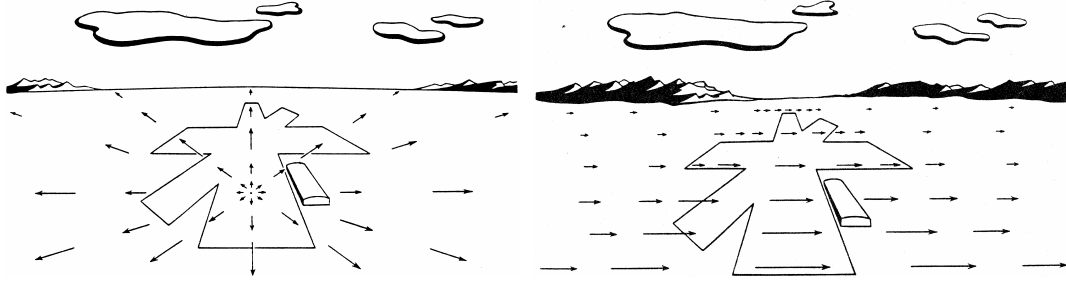
3 boyutlu noktalar uzayda hareket ederlerken, karşılık gelen görüntü noktaları da hareket eder. Hareket alanı, görüntüdeki tüm noktalar için görüntüde yer alan hareket vektörlerinden oluşmaktadır.

Hareket alanında yer alan hareket vektörlerinden göreceli hareket yönü ve hız çıkarımı yapılabilmektedir (bakınız Şekil 1.4).



Şekil 1.4 (a) İleri hareket, (b) dönme, (c) yatay çevrim, (d) yakın objeler daha hızlı hareket ediyor görünür.

Şekil 1.5 'te iki farklı hareket alanı örneği yer almaktadır; ilk örnekte (Şekil 1.5(a)) gözlemci ileriye doğru hareket ettiği için, hareket vektörleri her yönde görünmektedir. İkinci örnekte (Şekil 1.5(b)) ise gözlemci hareket ederken sağa bakmakta ve paralel hareket vektörleri elde edilmektedir.



Şekil 1.5 (a) Bir iniş alanında belirli bir noktaya inerken ileriye bakan bir pilotun hareket alanı. (b) Pilotun sağa bakarken gördüğü hareket alanı.

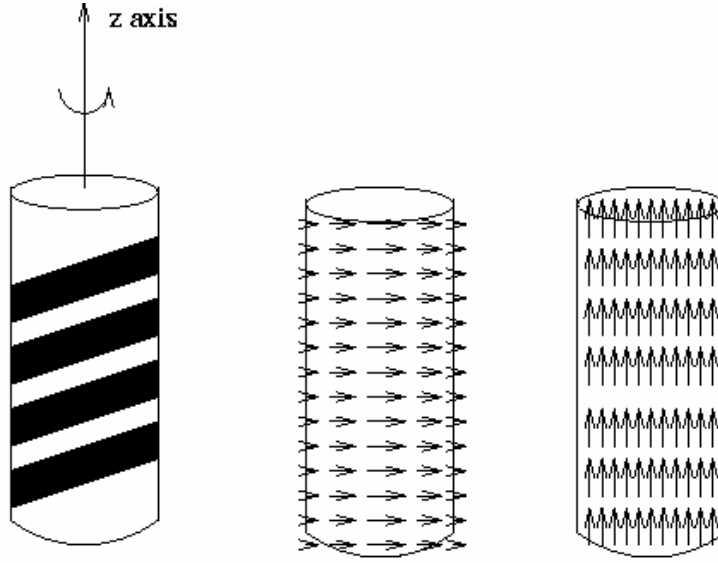
1.4 Optik Akış Tanımı

Canlılar üç boyutlu nesne hareketini retinalarına düşen ışık ile algılamakta, bilgisayarlar görmeye hareketi algılamak için görsel sensörlerin bir dizisi kullanılır. Örneğin CCD veya CMOS kameralar kullanılarak görüntünün her pikseline karşılık düşen şiddet genliği elde edilebilir. Fakat bu sensörler sayesinde sadece piksellerin düzlemsel konumu ve lineer şiddet bilgisi çıkarımı yapılabilir. Şiddet bilgisi sayesinde görüntü üzerindeki her noktaya bir değer atanabilir ve bu değer kullanılarak sonraki görüntü çerçevelerindeki hız tespiti yapılabilir. Görüntü üzerindeki her noktanın hızını bulabilmek için ise ek yaklaşımlar geliştirilmiştir. Bu yaklaşımlar, her noktanın görüntü çerçeveleri arasında ne kadar yer değiştirdiğini hesaplayarak hız tespiti yapmaktadır.

Burada bahsedilen yöntemlerin asıl amacı sensörlerden elde edilen Şiddet bilgisinden hız alanını elde edebilmektir. Fakat hız alanı bilgisini elde edebilmek için ardışık görüntü dizileri arasında her şiddet noktasının nereye hareket ettiği bulunmak zorundadır. Bununla birlikte, şiddet noktaları etiketlenemez ve direk olarak tanınmaz çünkü aynı şiddet değerine sahip bir çok komşu piksel olabilir.

3 boyutlu sahnede gözlemci hareket ettiğinde, görüntüde meydana gelen gözle görünür hareket, “Optik Akış” olarak adlandırılır. Optik akış, görüntüdeki hareketin yön ve hızını tanımlar. Genellikle optik akış hareket alanına karşılık gelir fakat bu denklik her durum için geçerli değildir. Örneğin, dönen direğin hareket alanı ve optik akışı birbirinden farklıdır

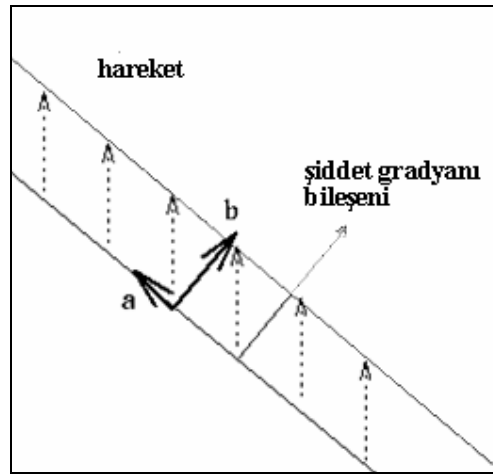
(bakınız Şekil 1.6). Genellikle, bu tür durumlara nadir rastlanır ve bu tez çalışmasında optik akışın hareket alanına karşılık geldiği varsayımı yapılmaktadır.



(a) Direk (b) Hareket Alanı (c) Optik Akış

Şekil 1.6 Direğin optik akışı ve hareket alanı.

Buradaki problem sadece şiddet gradyanı doğrultusundaki optik akış bileşeninin hesaplanabilmesidir; şiddet gradyanının tanjant bileşeni hesaplanamaz. Bu problem Şekil 1.6'da gösterilmektedir. Şiddet gradyanı; görüntü üzerindeki skalar alanda en büyük değişimin yönünü gösteren ve genliği en büyük değişim oranı olan vektör alanıdır.



Şekil 1.7 Belirsizlik problemi. Sadece b bileşeni hesaplanabilir.

Görüntü örnekleme oranının nesne hareketinden daha yavaş olduğu durumlarda ise, çerçeveler arasında hareket mesafesi çok büyük olacaktır ve piksel hareketini tanımlamak daha zor olacaktır. Bu durumlarda hareket tespiti yöntemleri uygulanamaz duruma gelir. Veri, örtüşmeden sakınmak için uygun bir şekilde örneklenmelidir; Nyquist örnekleme koşulları sağlanmalıdır. Bununla birlikte pratik uygulamalarda örnekleme oranı veya çerçeve hızı, şiddet noktalarının değişim hızından daha büyüktür ve bu değişim hızını takip etmek daha kolay olacağından hareket tespiti yöntemleri doğru sonuçlar üretecektir.

Optik akış tahmini; hedef izleme, model tanıma ve diğer görüntü işleme uygulamalarında kullanılmaktadır. Optik akış, robot biliminin yanı sıra video kodlama gibi farklı uygulamalarda da kullanılır. Video kodlamada, parlaklığı değişmeyen piksel bloklarının hareketi temel alınır. Robot biliminde ise çarpışmadan kaçma ve rota belirleme için daha karmaşık optik akış algoritmaları geliştirilmiştir.

1.4.1 İki Boyutlu Hareket Kısıtlama Denklemi

Hız alanının nedensel olarak sadece iki boyutlu şiddet bilgisinden tahmin edilmesi mümkün değildir. Nesne hareketinin görsel sensör tarafından algılanması sonucu iki boyutlu düzlem üzerinde meydana gelen görünür hız, optik akış olarak adlandırılır ve kullanılan yöntemler ile optik akış alanı, hız alanına yakınlştırılır (J.L.Barron and N.A.Thacker, 2004).

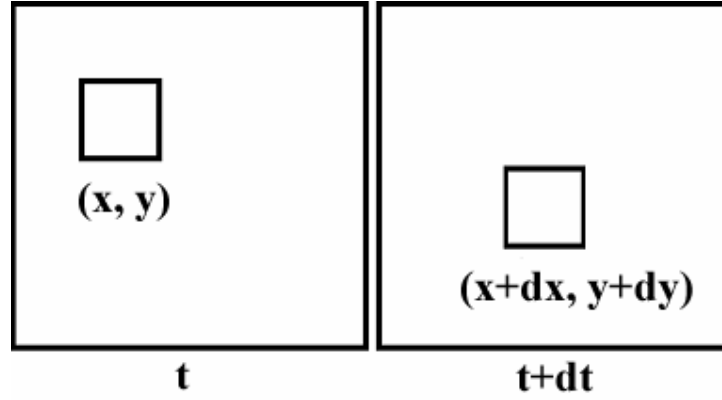
Görüntü üzerinde yer alan $r = (x, y)$ noktasındaki parlaklığın $I(x, y, t)$ olduğu varsayımı yapılmaktadır. Görüntüde belirli bir noktanın parlaklığı zamanla sabittir:

$$\frac{dI}{dt} = 0 \quad (1.7)$$

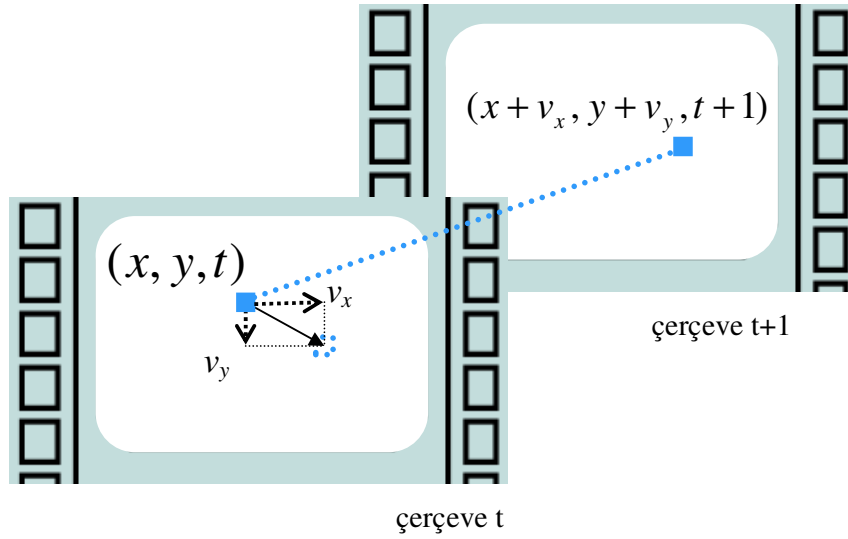
Ayrıca nesnenin parlaklığı önceki ve sonraki görüntü çerçevelerinde sabit kalacaktır:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (1.8)$$

Şiddet zamana göre değişiyorsa nesnenin hareket tespiti imkansızlaşır; pikselin bir sonraki görüntü dizisindeki yeri hesaplanamaz. Şiddet değişmezliği varsayımı, “Hareket Kısıtlama Denklemi”nin temellerini oluşturur. Bir noktanın şiddet değişmezliği Şekil 1.6 ’da gösterilmektedir.



Şekil 1.8 (x, y, t) konumundan, $(x + dx, y + dy, t + dt)$ konumuna hareket eden noktanın şiddet değişmezliği.



Şekil 1.9 Görüntüde bir noktanın yer değiştirmesinin zamana bağlı olarak hız bileşenleri ile gösterilmesi.

Parlaklığın uzay-zamansal düzlemde değişmediği kabulü ile, $r = (x, y)$ pikseli etrafındaki (dx, dy, dt) küçük değişimi için bir noktadaki parlaklığın Taylor açılımı aşağıdaki denklemle gösterilebilir:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + dx \frac{\partial I}{\partial x} + dy \frac{\partial I}{\partial y} + dt \frac{\partial I}{\partial t} + YDT \quad (1.9)$$

Taylor serileri açılımdan elde edilen denklemde $I(x, y, t) = I(x + dx, y + dy, t + dt)$

olduğundan bu terimler birbirini yok edecektir. Bununla birlikte, en az seviyede zamansal değişim kabulü yapıldığı için, $dt \rightarrow 0$, yüksek dereceden terimler (YDT) dikkate alınmaz.

(1.9) denklemi tekrar düzenlenirse aşağıdaki sonuç elde edilir:

$$\begin{aligned}\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt &= 0 \\ \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} \frac{dt}{dt} &= 0 \\ \frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} &= 0\end{aligned}\tag{1.10}$$

(1.10) denkleminin matris gösterimi (1.11) denklemi ile verilmiştir:

$$\frac{\partial I(x, y)}{\partial t} = - \left(\frac{dx}{dt} \frac{dy}{dt} \right) \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{pmatrix}\tag{1.11}$$

veya

$$I_t = -(v_x \ v_y) \begin{pmatrix} I_x \\ I_y \end{pmatrix}\tag{1.12}$$

burada $v_x = \frac{dx}{dt}$ ve $v_y = \frac{dy}{dt}$, görüntü hızının veya optik akışın x ve y bileşenleri; $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ ve

$\frac{\partial I}{\partial t}$, (x, y, t) noktasındaki görüntü şiddet türevleridir. Genelde bu kısmi türevler aşağıdaki şekilde kısa formda yazılmaktadır:

$$I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}, I_t = \frac{\partial I}{\partial t}\tag{1.13}$$

Optik akışın x ve y bileşenleri olan (v_x, v_y) ile şiddet türevleri olan (I_x, I_y, I_t) arasındaki farka dikkat edilmelidir.

$$(I_x, I_y) \cdot (v_x, v_y) = -I_t\tag{1.14}$$

veya

$$\nabla I \cdot \vec{v} = -I_t \quad (1.15)$$

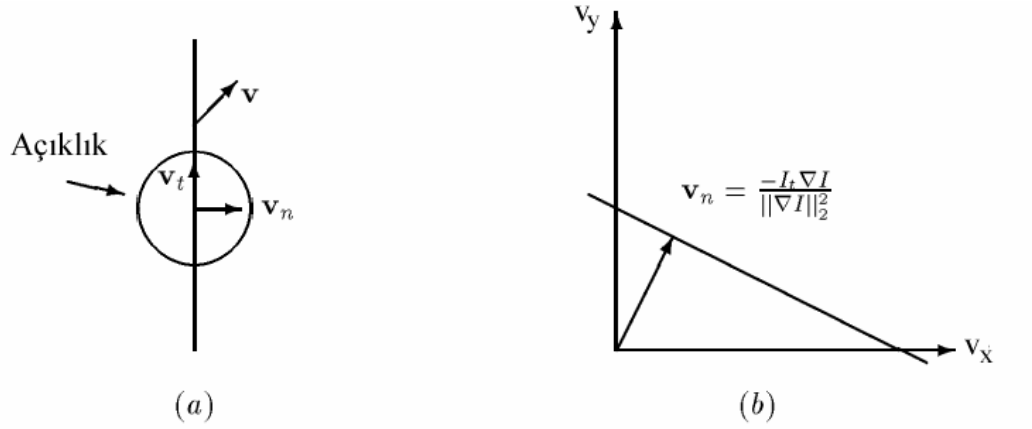
burada $\nabla I = (I_x, I_y)$, uzaysal şiddet gradyeni ve $\vec{v} = (v_x, v_y)$, (x, y) pikselinde t anındaki görüntü hızı veya optik akıştır. (1.15) denklemi “İki Boyutlu Hareket Kısıtlama Denklemi” olarak adlandırılmaktadır.

“İki Boyutlu Hareket Kısıtlama Denklemi” iki bilinmeyenli bir denklemdir ve bu denklemin çözümü sonucunda Şekil.1.10(b)’de gösterildiği gibi doğru denklemi elde edilir. Doğrultusu bu doğru üzerinde belirli bir (x, y) üzerinde olan bir çok hız bileşeni elde edilebilir. Bu nedenle sadece hareket kısıtlama denkleminde elde edilen doğru, hızın tam yerini bulmaya yeterli değildir ve bu problem “Belirsizlik Problemi” olarak adlandırılır. Yerel görüntü parlaklığı yapısından tam görüntü hızı elde edilemese de “Hareket Kısıtlama” doğrusuna normal gelen görüntü hızı hesaplanabilir.

Optik akış, ek bileşenler olmadan komşu noktalardan bağımsız olarak görüntü üzerinde bir noktada hesaplanamaz; çünkü her görüntü noktasındaki hız alanı iki bileşene sahipken harekete bağlı görüntü düzlemindeki bir noktanın görüntü parlaklığındaki değişim sadece bir bileşen sağlar.

Şekil 1.10(a)’da yukarı ve sağa ilerleyen bir doğru, dairesel belirsizlik boyunca resmedilmiştir. Bu doğrunun gerçek hızı $\vec{v} = (v_x, v_y)$ belirsizlik probleminden dolayı elde edilemez fakat “Hareket Kısıtlama” doğrusu üzerinde en küçük genliğe sahip normal hız, $\vec{v}_n$ hesaplanabilir. Bu durumda, sadece yerel şiddet yapısına normal gelen hızın bileşeni örneğin bir kenar elde edilebilir. Tam görüntü hızı hesabında, $\vec{v}_n$ normal hız bileşeninin yanı sıra $\vec{v}_t$ hızın teğetsel bilgisi de bulunmak zorundadır.

Sonuç olarak, optik akış yöntemleri ile tam görüntü hızını hesaplayabilmek için “Hareket Kısıtlama” denklemine ek olarak aynı bilinmeyenlere sahip ikinci bir denklemi elde etmeye çalışılmıştır.



Şekil 1.10 (a) Belirsizlik problemi, (b) Hareket Kısıtlama denkleminde normal gelen hız.

Normal hızın genliği ve doğrultusu ($\vec{v}_n = v_n \hat{n}$), şiddet türevleri I_x, I_y ve I_t 'nin yardımıyla hesaplanabilir:

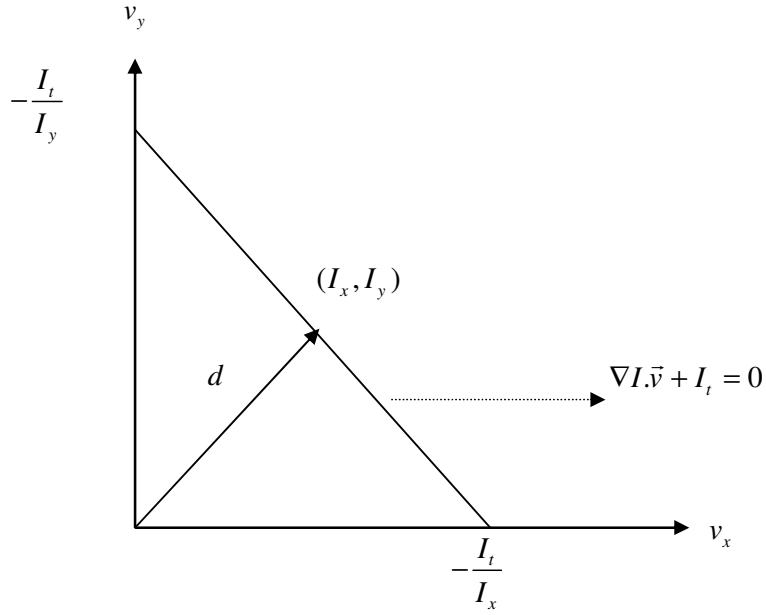
$$v_n = -\frac{I_t}{\|\nabla I\|_2} \text{ ve } \hat{n} = \frac{(I_x, I_y)}{\|\nabla I\|_2} \quad (1.16)$$

v_n ve $\hat{n}$, sırasıyla, birim normal hız genliği ve birim normal hız doğrultusudur. Normal hız vektörü, bu bileşenlerin noktasal çarpımıdır:

$$\vec{v}_n = v_n \hat{n} = \frac{-I_t (I_x, I_y)}{\|\nabla I\|_2^2} \quad (1.17)$$

burada $\vec{v}_n$, birim normal hız ve $\nabla I = (I_x, I_y)$, uzaysal şiddet gradyanıdır.

Optik akış yöntemleri; Lucas ve Kanade optik akış algoritması, normal hız hesabına izin verirken; Horn ve Schunck optik akış algoritması bu hesabı gerçekleştiremez.



Şekil 1.11 İki Boyutlu Hareket Kısıtlama Denkleminin grafiksel gösterimi.

İki boyutlu Hareket Kısıtlama denkleminde elde edilen doğrunun grafiksel gösterimi Şekil 1.11’de gösterilmiştir. Bu denklemden elde edilen x ve y yönündeki hız denklemleri (1.16) ve (1.17)’de hesaplanmıştır ve x ve y düzlemlerini kesim noktaları belirlenmiştir.

$$v_y = -\frac{I_x}{I_y}v_x - \frac{I_t}{I_y} \quad (1.18)$$

$$v_x = -\frac{I_y}{I_x}v_y - \frac{I_t}{I_x} \quad (1.19)$$

burada $d = \frac{-I_t}{\sqrt{I_x^2 + I_y^2}}$, kısıtlama denklemi üzerindeki belirli bir şiddet gradyanı doğrultusundaki optik akış bileşenidir.

Optik akış ve hız alanı birbirine bağlı olmasına karşın her durumda aynı olmayabilir. Optik akış ve hız alanı hesabındaki bu ince fark dikkate alınmalıdır. Gözlenen üç boyutlu görüntünün her noktasının konumu bilinemez fakat bu noktaların hareketi hesaplanarak her bir noktanın yer değiştirmesi bulunmaya çalışılır.

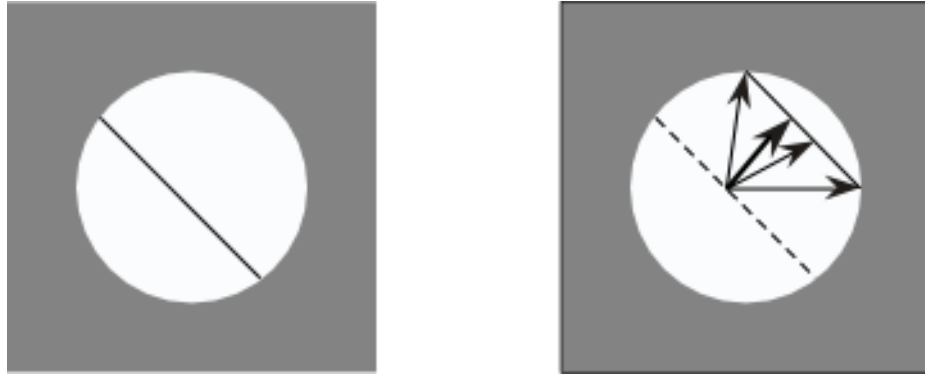
1.4.2 Belirsizlik Problemi (Aperture Problem)

Hız tahmini yöntemlerinde optik akış hız, alanına yaklaştırılarak görüntü hızı hesaplanmaya çalışılmıştır. Optik akış bileşeni şiddet gradyanı boyunca

$$d = -\frac{I_t}{\sqrt{I_x^2 + I_y^2}} \text{ şeklinde}$$

yazılabilir. Bu ifade $I_x = I_y = 0$ olduğu durumlarda belirlenemez ve dolayısıyla dokusuz veya gürültü seviyeleri ile karşılaştırılabilir dokuya sahip alanlarda optik akış doğru bir şekilde hesaplanamaz. Pratik uygulamalarda şiddet ölçümleri; elektronik, ısıl veya çevresel şiddet değişimlerine bağlı olarak gürültü edinebilir. Gürültü etkisi sonucunda parlaklığın uzaysal türevlerinin hata hesaplamalarının, $(I_x = I_y) \rightarrow 0 \Rightarrow I_x < dI_x, I_y < dI_y$ hesabından daha büyük olduğu durumlarda, doğru optik akış hesaplaması mümkün olmayacaktır.

Her görüntü noktasındaki hız alanı iki bileşene sahiptir fakat görüntü düzlemindeki bir noktada meydana gelen harekete bağlı görüntü parlaklığındaki değişim sadece bir bileşen verir. Bunun sonucu olarak optik akış, komşu noktalardan bağımsız olarak ek bileşenler ortaya konulmadan görüntü üzerindeki herhangi bir noktada hesaplanamaz. “Belirsizlik Problemi” olarak adlandırılan bu durum optik akışa bağlı önemli bir problemi işaret eder.

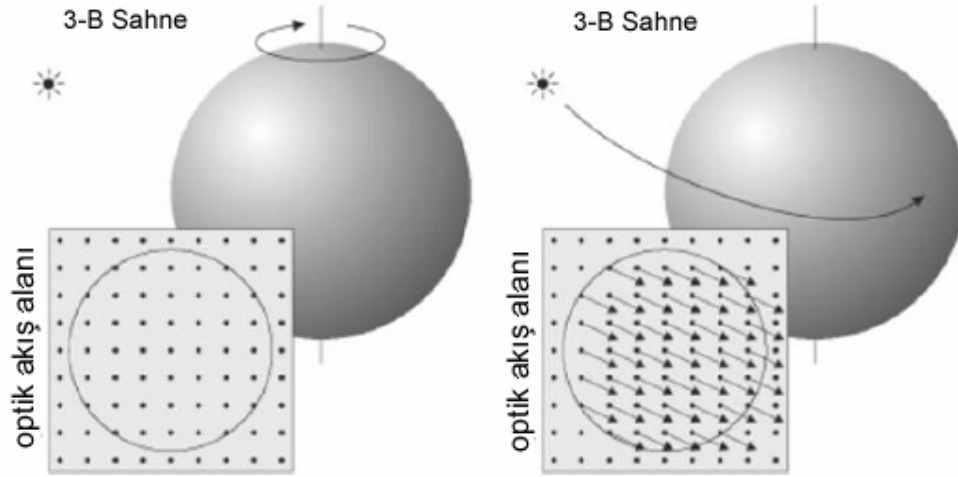


Şekil 1.12 (a) t anında küçük bir mesafe boyunca bir doğru gözlenmektedir. (b) (t+dt) anında, doğrunun her noktasının nereye hareket ettiğine tam olarak karar verilmesi. mümkün değildir.

Yerel görüntü ölçümlerinden akışın (hareketin) sadece doğruya dik bileşeni bulunabilir.

Literatürde görüntü hareketi hesabında bazı istisnalar mevcuttur. Örneğin sabit bir aydınlatma altında dokusuz bir kürenin dönmesi algılanamaz. Hız alanı statik olmasına karşın hesaplanan optik akış sıfır olacaktır (bakınız Şekil 1.13(a)). Benzer şekilde hareketli bir aydınlatma altında sabit dokusuz bir küre hareket ediyormuş gibi algılanır. Hız alanı sıfır olmasına karşılık optik akış statik bir değer alacaktır (bakınız Şekil 1.13(b)). Örneklerden de görüldüğü

üzere optik Akış, hız alanına her zaman iyi bir yaklaşım sağlamamaktadır. Optik akışı kullanan herhangi bir model, bu tür hataları telafi edecek başka bileşenlere sahip olmalıdır. Bunun yanı sıra optik akış temeline dayanan tüm yöntemler parlaklık hareketi tutarsızlığına bağlı olan durumlarda çalışırken başarısızlığa uğrayacaktır.



Şekil 1.13 (a) Sabit ışık kaynağına sahip dönen küre (b) Hareket eden ışık kaynağına sahip sabit küre.

Ayrıca, “Belirsizlik Problemi” ile baş edebilmek için sahnedeki nesnelerin katı ve şekil değişimi olmadığı varsayımı yapılır ve bu sayede büyüme, küçülme, deformasyon ve/veya kırılmalar dikkate alınmaz. Optik akış yöntemleri bu nedenlerle nesnelerin hareketleri süresince enerji ve şekillerinin değişmezliği kabulünü yapar. Genellikle optik akış hesaplamalarında daha doğru sonuçlar elde edebilmek için uygulamaya özellik tanıma metodu da dahil edilir.

Optik akış hesabında karşılaşılan diğer problemler arasında görüntünün şiddet değişmezliğinin bozulması, görüntüdeki gölgeler, yerel şiddet düzgünlüğünü bozan yüzey saydamlığı yer almaktadır. Bunların yanı sıra, hareket eden veya sabit nesnelerin örtülmesi doğru tahmini engelleyecektir. Ayrıca bir camın önünde veya yansıyan yüzeyler üzerinde hareket etme, hız alanı yaklaşımına tutarsız bir optik üretecektir.

Tüm bu problemlere rağmen, doğal sahnelerde optik akış hesabı genellikle yeterli derecede çalışmaya ayarlanabilir. Bir sonraki bölümde bazı optik akış tekniklerinden bahsedilmiştir.

1.5 Optik Akış Teknikleri

Görüntü dizilerinin işlenmesindeki temel problem optik akışın veya görüntü hızının ölçülme zorluğudur. Bu bölümde anlatılan yöntemlerin genel amacı, iki boyutlu hareket alanına yaklaşık bir hesaplama yapmaktır (J.L. Barron, D.J. Fleet ve S.S. Beauchemin, 1994).

Optik akış hesaplaması için birçok teknik ileri sürülmüştür ve günümüzde hala diğerleri ortaya çıkmaya devam etmektedir. Optik akış teknikleri aşağıda bahsedilen üç gruptan birine bağlı olarak sınıflandırılabilir:

1. Fark Teknikleri: uzay-zamansal şiddet türevlerinden görüntü hızını hesaplar.
2. Frekans-temelli Teknikler: hız ayarlı filtre çıkışındaki enerji/faz bilgisini kullanırlar.
3. Eşleştirme Teknikleri: az sayıda görüntüden (genellikle iki veya üç görüntü dizisinden) değişik görüntü özelliklerini eşleştirerek görüntü yer değişimlerini hesaplar.

Bu üç grup yaklaşım arasında uygulama ve performans farklılıkları olmasına karşın genelde bir çok bakımdan denk olarak düşünülürler. Genel yapısı bakımından bu teknikler üç işleme aşamasına dayanarak incelenebilir:

1. Görüntü üzerinde ilgilenilen işaret yapısını elde edebilmek ve işaret/gürültü oranını arttırmak için alçak-geçiren veya bant-geçiren filtre kullanarak ön-filtreleme veya yumuşatma,
2. Hızın normal bileşenlerini hesaplamak için temel ölçümlerin hesaplanması; örneğin uzay-zamansal türevlerin elde edilmesi,
3. İki boyutlu akış alanını üretebilmek için ön filtreleme ve temel hesaplamaların entegrasyonunun yapılması.

Burada fark yöntemleri arasında yer alan genel ve yerel yaklaşımlar üzerinde durulmuştur. Bu tez çalışmasında genel fark tekniği temeline dayanan Horn ve Schunck optik akış modeli ve yerel fark tekniği temeline dayanan Lucas ve Kanade optik akış modeli incelenmiştir.

2. FARK TEKNİKLERİ

2.1. Fark Tekniklerinin Genel Yapısı

Optik akış, görüntü dizilerindeki yerel türevler üzerine kurulu yerel görüntü hareketi yaklaşımıdır. İki boyutlu görüntüde yerel türevler hesaplanarak her görüntü pikselinin ardışık görüntüler arasında ne kadar ilerlediği belirlenir. Optik akış, gözlenen görüntülerin uzaysal düzenlemesi ve bu düzenlemenin değişim oranı hakkında yararlı bilgiler vermektedir. Hareket eden görüntüler görüntü parlaklığında zamansal değişimlere neden olmaktadır ve tüm bu zamansal şiddet değişimlerinin sadece harekete bağlı olduğu varsayımı yapılmaktadır (B.K.P. Horn ve B.G. Schunck, 1981).

Genellikle iki boyutlu türev hesaplamaları, alçak ve yüksek geçiren filtrelerin tekrarlanan uygulamaları ile gerçekleştirilmektedir. Diferansiyel optik akış hesabı aşağıdaki şekilde iki temel adımda anlatılır:

1. Uzay-zamansal şiddet türevlerini hesaplama,
2. Normal hız hesabını doğru hız tahminine yaklaştırma. En küçük kareler yaklaşımı ile yerel yöntemler kullanılarak hesaplama (örneğin Lucas ve Kanade yöntemi) veya düzenleme ile genel yöntemler kullanılarak hesaplama (örneğin Horn ve Schunck yöntemi).

2.1.1 Fark Alma

Hangi optik akış metodu kullanılırsa kullanılsın, her zaman görüntü şiddet türevleri hesaplanmak zorundadır. I_x , I_y ve I_t şiddet türevlerinin sürekli olması hız tahmininde önemlidir. Bu nedenle, türev hesaplamaları yapılırken hepsi aynı zamanda görüntü üzerindeki aynı noktaya işaret etmelidir.

Optik akış hesabında yaklaşık fark hesabı için bir çok yöntem kullanılmaktadır. Burada bahsedilen fark hesabı yöntemi, sekiz ardışık ölçüme sahip bir küpün merkez noktasındaki I_x , I_y ve I_t türev tahminlerini veren bir set kullanmaktadır.

Sekiz ardışık ölçüm arasındaki uzay ve zamansal ilişki Şekil 2.1’de gösterilmektedir. Küpün merkezindeki görüntü parlaklığının üç kısmi türevi, küpün dört paralel kenarı boyunca alınan farkların ortalamasından tahmin edilir. Kolon j indeksi x doğrultusuna, satır i indeksi y doğrultusuna karşılık gelir ve k indeksi zaman doğrultusunda ilerlemektedir.

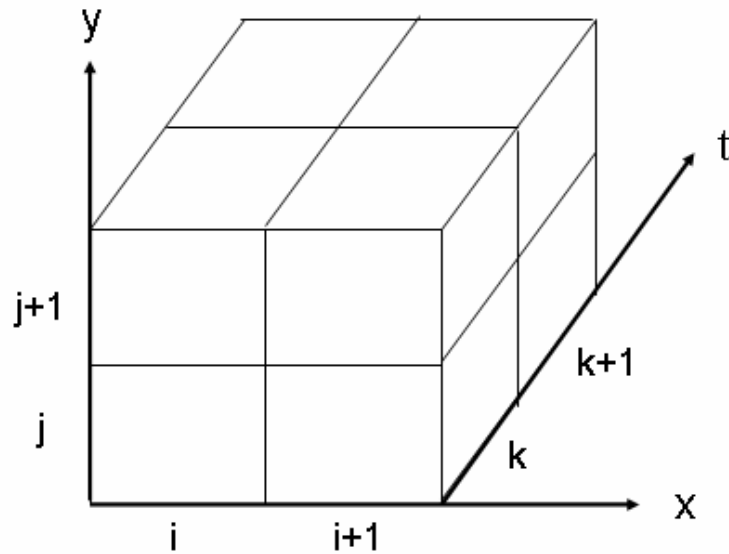
Hesaplanan türev tahminlerinin her biri, küpteki bitişik ölçümlerden alınan dört farkın ortalamalarıdır:

$$\begin{aligned}
 I_x &\approx \frac{1}{4}(I_{i,j+1,k} - I_{i,j,k} + I_{i+1,j+1,k} - I_{i+1,j,k} \dots \\
 &\quad + I_{i,j+1,k+1} - I_{i,j,k+1} + I_{i+1,j+1,k+1} - I_{i+1,j,k+1}) \\
 I_y &\approx \frac{1}{4}(I_{i+1,j,k} - I_{i,j,k} + I_{i+1,j+1,k} - I_{i,j+1,k} \dots \\
 &\quad + I_{i+1,j,k+1} - I_{i,j,k+1} + I_{i+1,j+1,k+1} - I_{i,j+1,k+1}) \\
 I_t &\approx \frac{1}{4}(I_{i,j,k+1} - I_{i,j,k} + I_{i+1,j,k+1} - I_{i+1,j,k} \dots \\
 &\quad + I_{i,j+1,k+1} - I_{i,j+1,k} + I_{i+1,j+1,k+1} - I_{i+1,j+1,k})
 \end{aligned} \tag{2.1}$$

Görüntünün uzay ve zamansal türevlerinin maskeleri aşağıdaki şekilde yazılmaktadır:

$$\begin{aligned}
 M_x &= \frac{1}{4} \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} & M_y &= \frac{1}{4} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} & M_t &= \frac{1}{4} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \\
 I_x &= M_x * (I_1 + I_2) & I_y &= M_y * (I_1 + I_2) & I_t &= M_t * (I_2 - I_1)
 \end{aligned} \tag{2.2}$$

Burada uzunluk birimi, her görüntü çerçevesindeki grid mesafesidir; zaman birimi, görüntü çerçeveleri arasındaki periyottur.



Şekil 2.1 Türev hesaplamalarında kullanılan sekiz ardışık ölçüme sahip küp.

2.1.2 Akış Hızı Laplasiyenleri Hesabı

Optik akış yöntemlerinde v_x ve v_y hız bileşenlerinin Laplasiyenleri de hız tahmininde kullanılmaktadır. Laplasiyen hesabında kullanılan güvenilir bir yaklaşım aşağıdaki denklemle verilmektedir:

$$\nabla^2 v_x \approx \kappa(\bar{v}_{x(i,j,k)} - v_{x(i,j,k)}) \text{ ve } \nabla^2 v_y \approx \kappa(\bar{v}_{y(i,j,k)} - v_{y(i,j,k)}) \quad (2.3)$$

burada yer alan yerel ortalamalar, $\bar{v}_x$ ve $\bar{v}_y$, (2.4) denkleminde belirtilmektedir:

$$\begin{aligned} \bar{v}_{x(i,j,k)} = & \frac{1}{6} \{ v_{x(i-1,j,k)} + v_{x(i,j+1,k)} + v_{x(i+1,j,k)} + v_{x(i,j-1,k)} \} \dots \\ & + \frac{1}{12} \{ v_{x(i-1,j-1,k)} + v_{x(i-1,j+1,k)} + v_{x(i+1,j+1,k)} + v_{x(i+1,j-1,k)} \} \\ \bar{v}_{y(i,j,k)} = & \frac{1}{6} \{ v_{y(i-1,j,k)} + v_{y(i,j+1,k)} + v_{y(i+1,j,k)} + v_{y(i,j-1,k)} \} \dots \\ & + \frac{1}{12} \{ v_{y(i-1,j-1,k)} + v_{y(i-1,j+1,k)} + v_{y(i+1,j+1,k)} + v_{y(i+1,j-1,k)} \} \end{aligned} \quad (2.4)$$

Hız bileşenlerinin Laplasiyen hesabı, komşu piksellerin ağırlıklı ortalamalarından bir noktadaki değerin çıkarılması ile elde edilir. Şekil 2.2’de komşu piksellerin bölünebileceği uygun ağırlıkları göstermektedir.

$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{12}$
$\frac{1}{6}$	-1	$\frac{1}{6}$
$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{12}$

Şekil 2.2 Laplasiyen hesabında kullanılan ağırlıklar.

2.2 Horn ve Schunck Optik Akış Modeli

2.2.1 Yumuşatma Bileşeni

Daha önceki bölümlerde elde edilen Hareket Kısıtlama denklemi hız tahmininde yetersiz kalmaktadır. Fakat şiddet örüntüsünün her noktası birbirinden bağımsız olarak hareket edebiliyorsa, hala hız tahmini iyileştirmesi yapılabilme olasılığı bulunmaktadır. Bu durumda nesne üzerinde bulunan komşu noktalar benzer hızlara sahiptir ve görüntü üzerindeki şiddet örüntülerinin hız alanı neredeyse her yerde değişmektedir. Bununla birlikte, bir nesne diğer bir nesneyi kapattığında akışta süreksizlik gözlemlenir (B.K.P. Horn ve B.G. Schunck, 1981).

Hareket Kısıtlama denklemine getirilen ilave bileşeni tanımlananın bir yolu, bir noktadaki akış hızı ile noktayı da içeren küçük bir komşuluk üzerindeki ortalama arasındaki farkı sınırlamaktır. Eşdeğer olarak, bu sınırlamayı karşılamak için akışın x ve y bileşenlerinin toplam karesel Laplasiyenleri minimize edilmektedir. x ve y yönündeki v_x ve v_y hız bileşenlerinin Laplasiyenleri aşağıdaki denklemlerle ifade edilmektedir:

$$\begin{aligned}\nabla^2 v_x &= \frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2} \\ \nabla^2 v_y &= \frac{\partial^2 v_y}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2}\end{aligned}\tag{2.5}$$

Gözlemci düz bir nesneye paralel olarak değişirse, yüzeye dik olarak hareket eder ve sonrasında v_x ve v_y 'nin ikinci kısmi türevlerinin her ikisi de kaybolur. Çünkü gerçek hız değeri ile gözle görünür hız arasında herhangi bir sapma yoktur. Bu tür basit durumlarda, her iki denklemden elde edilen Laplasiyen sıfırdır.

2.2.2 Horn ve Schunck Yönteminde Hataları Minimize Etme

Görüntü üzerindeki bir noktanın parlaklığı diğer görüntü çerçevelerinde aynı kalmaktadır. Fakat gürültü nedeni ile Hareket Kısıtlama denklemi sıfıra eşit olmayabilir. Ayrıca, hız akışı tahmininde elde edilen hızlar gerçek hız değerinden sapmış olabilir.

Bu hataları iki şekilde ele alabiliriz:

1. Görüntü parlaklığının değişim miktarı için denklemdeki hataların toplamını minimize etme:

$$E_b = I_x v_x + I_y v_y + I_t \quad (2.6)$$

2. Hız akışında yumuşatma için sapma tahminini minimize etme:

$$E_c^2 = (\bar{v}_x - v_x)^2 + (\bar{v}_y - v_y)^2 \quad (2.7)$$

Bu iki denklemdeki hata faktörlerini birbirine bağlayan ağırlık ne olmalıdır? Pratikte görsel sensörlerin algıladığı görüntü şiddet bilgisi, kuantalama hatası ve gürültü tarafından bozulacaktır ve bu nedenle E_b hata değerinin idealde sıfır olması beklenmez. Hız akışındaki sapma ve görüntü parlaklığındaki değişim miktarı arasındaki α^2 ağırlık faktörü ile gösterilmektedir. Şiddet bilgisinin gürültüye olan eğilimi bu faktörün belirlenmesinde önemli bir etkidir.

Sonuç olarak minimize edilmek istenen toplam hata (2.8) denkleminde verilmektedir:

$$E^2 = \alpha^2 E_c^2 + E_b^2 \quad (2.8)$$

Horn ve Schunck yöntemi, tahmin edilen hız alanı $\vec{v} = (v_x, v_y)$ 'yi sınırlamak için genel yumuşatma terimi ile gradyan kısıtlama denklemini birleştirmekte ve aşağıdaki denklemi minimize etmektedir:

$$E^2 = \int_D (\nabla I \cdot \vec{v} + I_t)^2 + \alpha^2 \left[\left(\frac{\partial v_x}{\partial x} \right)^2 + \left(\frac{\partial v_x}{\partial y} \right)^2 + \left(\frac{\partial v_y}{\partial x} \right)^2 + \left(\frac{\partial v_y}{\partial y} \right)^2 \right] dx dy \quad (2.9)$$

D alanı görüntü boyunca tanımlanmıştır; α katsayısının genliği yumuşatma teriminin ilgili etkisini yansıtır.

2.2.3 Tekrarlama Yöntemi

Görüntü parlaklığında ve hız tahmininde karşılaşılan hataları azaltma, optik akış hız bileşenlerine (v_x, v_y) uygun değerler bulunarak gerçekleştirilmelidir. Hata denkleminin kısmi türevi alınarak aşağıdaki (2.10) denklem elde edilir:

$$\begin{aligned}\frac{\partial E^2}{\partial v_x} &= -2\alpha^2(\bar{v}_x - v_x) + 2(I_x v_x + I_y v_y + I_t)I_x \\ \frac{\partial E^2}{\partial v_y} &= -2\alpha^2(\bar{v}_y - v_y) + 2(I_x v_x + I_y v_y + I_t)I_y\end{aligned}\quad (2.10)$$

Bu iki hata türevi sıfıra eşitlenerek v_x ve v_y bileşenlerine bağlı iki denklem elde edilir:

$$\begin{aligned}(\alpha^2 + I_x^2)v_x + I_x I_y v_y &= (\alpha^2 \bar{v}_x - I_x I_t) \\ I_x I_y v_x + (\alpha^2 + I_y^2)v_y &= (\alpha^2 \bar{v}_y - I_y I_t)\end{aligned}\quad (2.11)$$

Katsayı matrisinin determinanı, $\alpha^2(\alpha^2 + I_x^2 + I_y^2)$ 'a eşittir. (2.11) denklemleri v_x ve v_y için çözümlenerek aşağıdaki denklemler elde edilir:

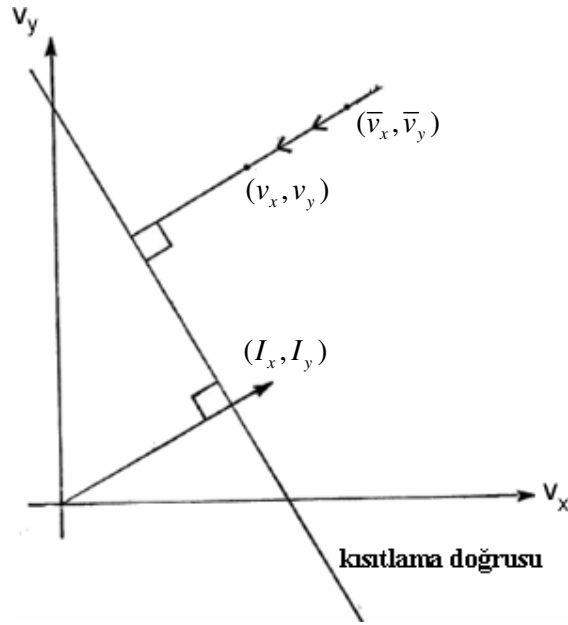
$$\begin{aligned}(\alpha^2 + I_x^2 + I_y^2)v_x &= +(\alpha^2 + I_y^2)\bar{v}_x - I_x I_y \bar{v}_y - I_x I_t \\ (\alpha^2 + I_x^2 + I_y^2)v_y &= -I_x I_y \bar{v}_x + (\alpha^2 + I_x^2)\bar{v}_y - I_y I_t\end{aligned}\quad (2.12)$$

Bu eşitlikler alternatif bir formda (2.13) denklemindeki şekilde tekrar yazılır:

$$\begin{aligned}(\alpha^2 + I_x^2 + I_y^2)(v_x - \bar{v}_x) &= -I_x[I_x \bar{v}_x + I_y \bar{v}_y + I_t] \\ (\alpha^2 + I_x^2 + I_y^2)(v_y - \bar{v}_y) &= -I_y[I_x \bar{v}_x + I_y \bar{v}_y + I_t]\end{aligned}\quad (2.13)$$

Hatayı indirgeyen akış hızı değerinin (v_x ve v_y), kısıtlama doğrusunu karşı sağ açılarda kesen doğru boyunca, kısıtlama doğrusuna karşı gelen doğrultuda bulunduğunu gösterir. Bu ilişkinin doğru üzerindeki gösterimi Şekil 2.3'de verilmektedir.

Sonuç olarak şiddet gradyanının küçük olduğu yerlerde α^2 ağırlık faktörünün önemli bir rolü olduğu görülmektedir. Bu parametre kabaca gradyanların kareleri toplamının $I_x^2 + I_y^2$ tahmininde beklenen gürültü değerine eşittir.



Şekil 2.3 Kısıtlama doğrusuna dik olarak akış hızının lokal ortalamasından çizilen doğruya bulunan, hatayı indirgeyen akış hızı değeri.

Hatalar toplamını en aza indirmek için tekrarlama denklemleri kullanılmaktadır. Horn ve Schunck yönteminde, Euler-Lagrange denklemlerini çözen Gauss-Seidel tekrarlama denklemleri ile hız akışı tespit edilmektedir:

$$\begin{aligned} v_x^{k+1} &= \bar{v}_x^k - \frac{I_x [I_x \bar{v}_x^k + I_y \bar{v}_y^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \\ v_y^{k+1} &= \bar{v}_y^k - \frac{I_y [I_x \bar{v}_x^k + I_y \bar{v}_y^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \end{aligned} \quad (2.14)$$

burada k , tekrarlama sayısını; v_x^0 ve v_y^0 başlangıç hız tahminlerini (tipik olarak sıfıra ayarlanır); $\bar{v}_x$ ve $\bar{v}_y$ ise v_x^k ve v_y^k 'nin komşuluk ortalamalarını gösterir.

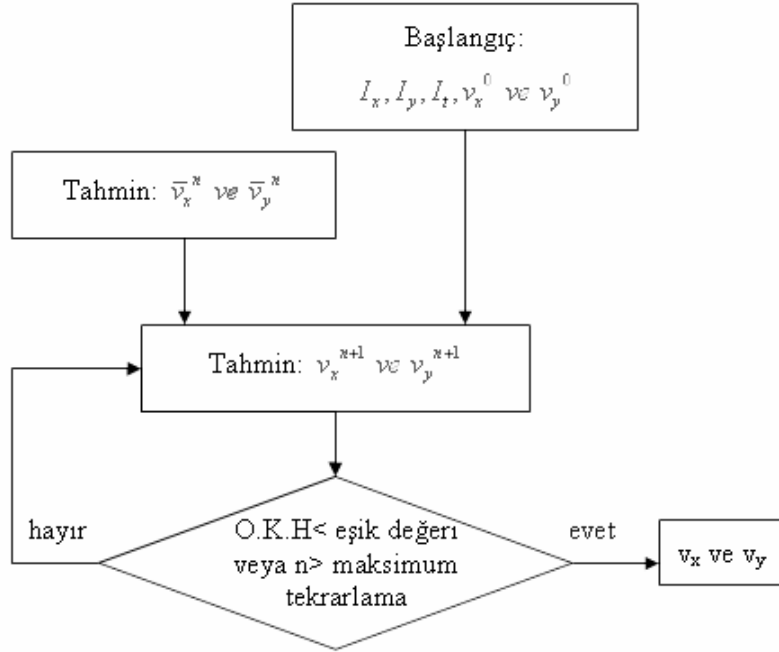
Yumuşatma faktörü α , pozitif bir sabittir. İki görüntü veya video dizisi arasındaki ilişkili hareket büyükse, yumuşatma faktörü için büyük bir pozitif skalar değer verilmelidir. Eğer ilişkili hareket küçükse, küçük bir pozitif skalar değer verilmelidir. Uygulamaya en iyi uyan yumuşatma faktörünü bulunması için birden fazla değer denenmesi gerekmektedir.

Ortalama hız ve elde edilen hız farkı belirli bir eşik değerinin altına indiğinde veya belirlenen bir tekrarlama sayısına ulaşıldığında, Gauss-Seidel tekrarlama durdurulmaktadır. Hem

eşik değeri hem de tekrarlama sayısı kullanıldığı durumda ise bu koşullardan biri sağlandığında tekrarlama durur. Eşik değeri, hız tahmininde kabul edilebilir sapma miktarını belirlemektedir. Tekrarlama sayısı ise, hareket eden en büyük nesnenin kesit alanından daha büyük olmalıdır. Nesne kesitinin önceden bilinmediği durumlarda ise, tüm görüntünün kesitinin kullanılması uygun olacaktır. Birkaç farklı tekrarlama sayısı kullanılarak elde edilen sonuçlar karşılaştırıldığında, belli bir tekrarlama sayısından sonra hata değerinde küçük değişiklikler meydana geldiği görülmektedir. Bu karşılaştırmalar sayesinde daha doğru bir tekrarlama sayısı elde edilebilir.

2.2.4 Horn ve Schunck Algoritması ile MATLAB Uygulaması

Genel fark yöntemlerinden Horn ve Schunck algoritmasının blok şeması Şekil 2.4'te gösterilmektedir. Blok şemadan da görüleceği üzere öncelikle başlangıç hız değerleri atanır ve uzay ve zamana ilişkin şiddet türevleri hesaplanır. Burada kullanılan başlangıç hız değerleri sıfırdır. Diğer bir yandan verilen iyi bir başlangıç seviyesi, tekrarlama tek bir adıma düşürebilir. Başlangıç hız değerlerine iyi bir tahmin olarak belirli bir çerçevede yer alan komşu piksellerin ortalamaları kullanılabilir. Tekrarlama denklemi sonucunda elde edilen hız değerleri bir sonraki tekrarlama kullanılmak üzere girilse verilir. Sonuç olarak, eğer elde edilen hız değerleri farkı belirli bir eşğin altında veya daha önceden belirlenen maksimum tekrarlama sayısının üzerinde ise tekrarlama durur.



Şekil 2.4 Horn ve Schunck yöntemi ile optik akış hesabı blok şeması.

2.2.4.1 Test Sonuçları

Horn ve Schunck fark tekniği kullanılarak MATLAB uygulaması gerçekleştirilmiştir. Girişe verilen iki görüntü çerçevesi için Hamburg Taksi orijinal görüntü dizileri kullanılmıştır. Kullanılan iki görüntü çerçevesinden elde edilen uzay ve zaman türevleri Şekil 2.6 (a), (b) ve (c)'de yer almaktadır. Birkaç α faktörü uygulanarak en uygun ağırlık faktörü bulunmuştur. Tekrarlama denklemleri sonucunda elde edilen hız bileşenleri MATLAB fonksiyonu quiver ile çizilerek Şekil 2.7 (a)'da resmedilmiştir. $v_x^2 + v_y^2$ hesaplanarak elde edilen hareketin enerji yoğunluğu Şekil 2.7 (b)'de gösterilmektedir. Cadde sahnesindeki küçük hareketler belirlenen ağırlık faktörü nedeni ile eşik değerinin altında kaldığı için tespit edilememiştir.

Hamburg Taksi Dizisi (256x191 piksel): Bir cadde sahnesinde hareket eden dört obje vardır: 1) köşeyi dönen taksi; 2) soldan sağa doğru ilerleyen, sol aşağıdaki araba; 3) sağdan sola ilerleyen, sağ alttaki kamyonet; ve 4) sol üstte yer alan yaya. Bu dört hareket eden objenin yaklaşık hızları sırasıyla; 1.0, 3.0, 3.0 ve 0.3 piksel/çerçeve'dir.

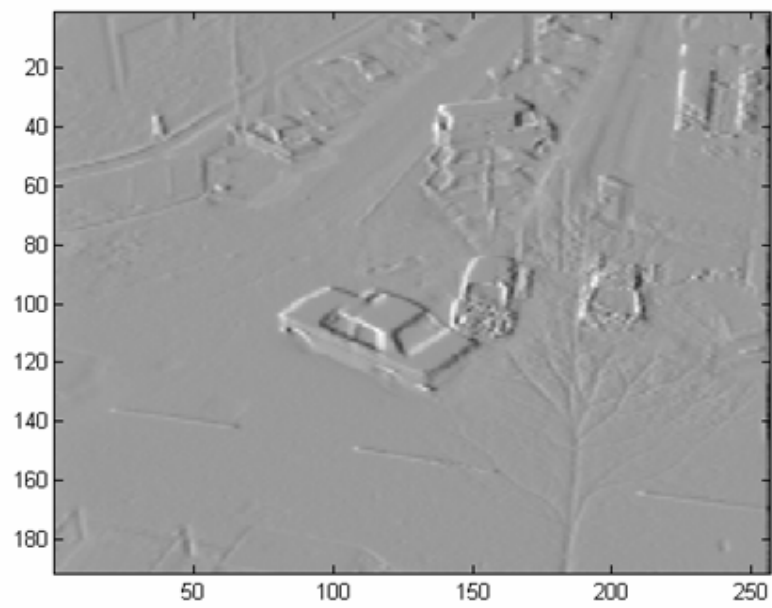


(a)

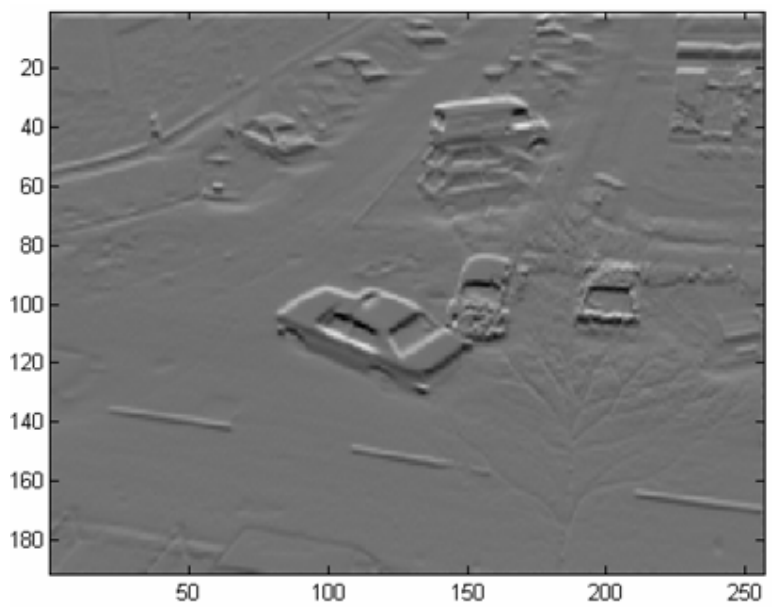


(b)

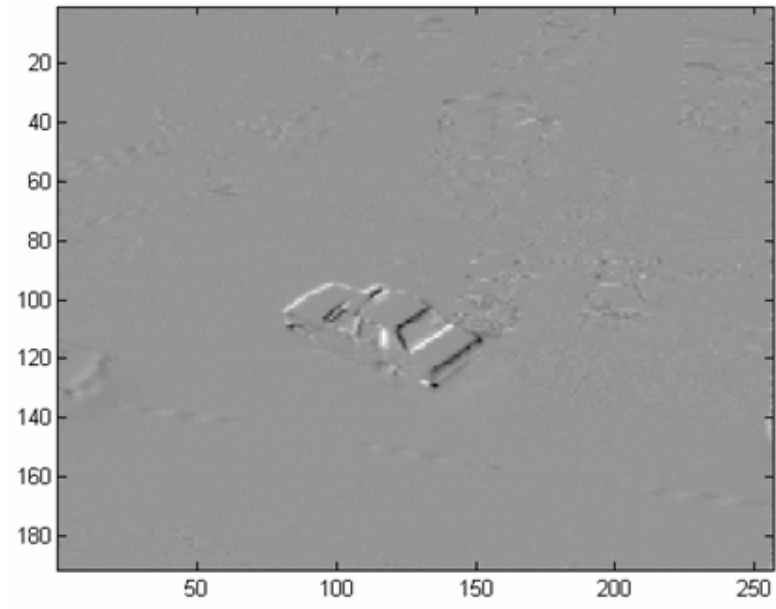
Şekil 2.5 (a), (b) Orijinal görüntü dizileri.



(a)

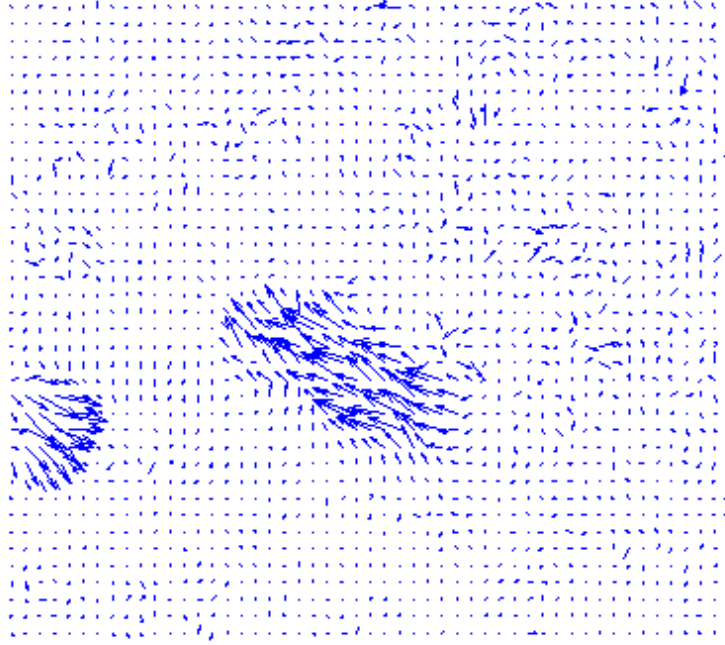


(b)

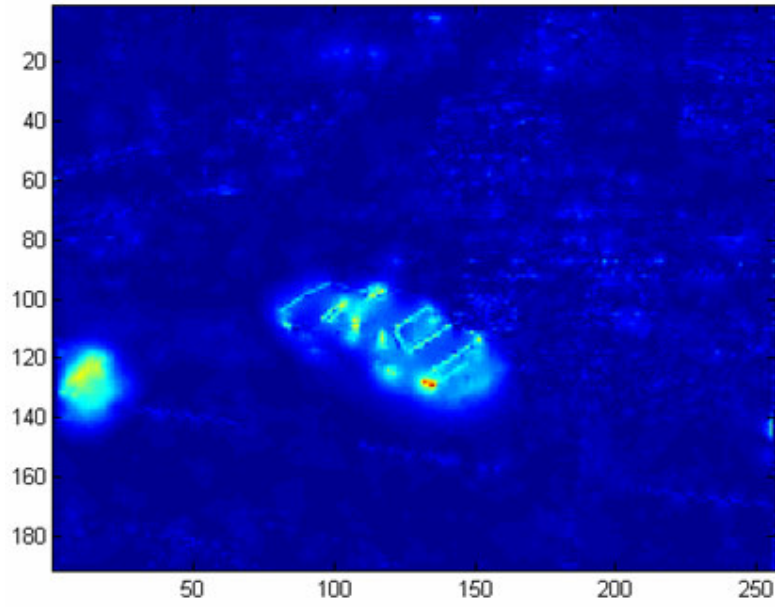


(c)

Şekil 2.6 (a) x yönündeki kısmi türev, (b) y yönündeki kısmi türev, (c) t yönündeki kısmi türev.



(a)



(b)

Şekil 2.7 (a) Optik akış alanı, (b) enerji alanı.

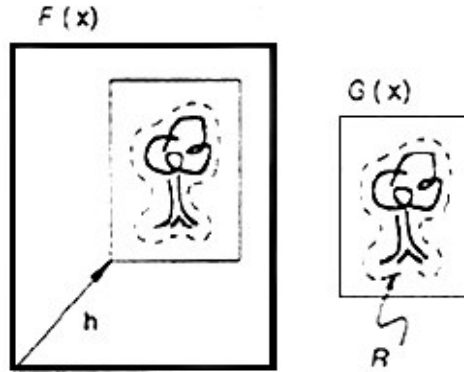
2.3 Lucas ve Kanade Optik Akış Modeli

Bu bölümde yerel fark tekniklerinden biri olan Lucas ve Kanade yöntemi incelenmektedir. Günümüzde Lucas-Kanade yöntemi, optik akış tahmini olarak da adlandırılan hız tahmininde iki görüntü dizisi kullanan farksal tekniklerinden biri olarak kullanılmaya devam etmektedir (B.D. Lucas ve T. Kanade, 1981).

2.3.1 Kaydetme Problemi

Bilgisayarla görmede, aynı sahneyi veya nesneyi farklı zamanlarda veya perspektiflerde ve farklı koordinat sistemlerinde örnekleyerek veri setleri oluşturulur. Görüntü kaydetme, farklı veri setlerini tek bir koordinat sistemine toplama işlemidir. Farklı ölçümlerden alınan verilerin karşılaştırması veya birleştirilmesi için kayıt gereklidir.

Görüntü dizilerinde karşılaşılan kaydetme problemi örnekle şu şekilde tanımlanmaktadır; x 'in vektör olarak belirlendiği iki ardışık görüntü dizisi olarak $F(x)$ ve $G(x)$ fonksiyonları verilir. R ilgilenilen alan içerisindeki x için $F(x+h)$ ve $G(x)$ arasındaki fark hesabını küçülten fark vektörü h bulunmaya çalışılmaktadır (bakınız Şekil 2.8).



(a)



(b)

Şekil 2.8 (a), (b) Görüntü kaydetme problemi

$F(x+h)$ ve $G(x)$ arasındaki tipik fark hesaplamaları şu şekildedir:

$$\begin{aligned} L_1 \text{ normu} &= \sum_{x \in R} |F(x+h) - G(x)| \\ L_2 \text{ normu} &= \left(\sum_{x \in R} [F(x+h) - G(x)]^2 \right)^{1/2} \end{aligned} \quad (2.15)$$

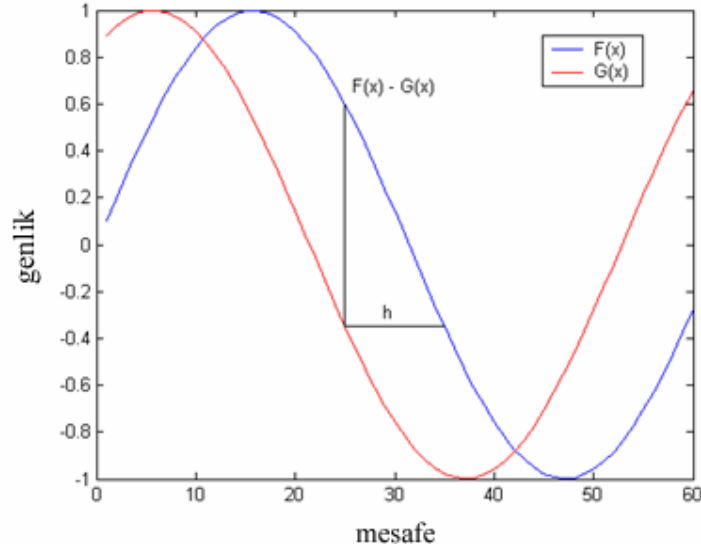
L_1 normu, L_2 normuna basit bir yaklaşım olarak kullanımı açısından kolaylık sağlamaktadır.

2.3.2 Kaydetme Algoritması

Öncelikle bir boyutlu kaydetme problemi için bir yaklaşım geliştirilmektedir. Daha sonra çok boyutlu görüntüler için tek boyutlu durumdan hareketle alternatif bir çözüm sunulmaktadır.

2.3.2.1 Tek Boyutlu Durum

Tek boyutlu kaydetme probleminde, iki eğri arasındaki ($F(x)$ ve $G(x)=F(x+h)$) yatay fark vektörü h bulunmak istenmektedir. (bakınız Şekil 2.9).



Şekil 2.9 Eşleştirilen iki eğri.

Bu problemin çözümü, x komşuluğundaki $F(x)$ 'in davranışına olan lineer yaklaşıma dayanmaktadır.

Küçük bir h değişimi için ayrık zamanlı türev (2.16) yaklaşıklığı ile hesaplanmaktadır:

$$\begin{aligned} F'(x) &\approx \frac{F(x+h) - F(x)}{h} \\ &= \frac{G(x) - F(x)}{h} \end{aligned} \quad (2.16)$$

ve buradan h değeri yaklaşıklığın sol tarafına alınırsa:

$$h \approx \frac{G(x) - F(x)}{F'(x)} \quad (2.17)$$

elde edilir. Bu algoritmanın başarılı olabilmesi için, bu yaklaşımın yeterli olmasını sağlayacak kadar h değerinin küçük olması istenmektedir.

(2.17) yaklaşıklığında verilen h yaklaşımı, x bileşenine bağlıdır. x bileşeninin değişik değerlerindeki h değerleri tahminlerini birleştirmek için basit bir şekilde ortalamaları alınmaktadır:

$$h \approx \sum_x \frac{G(x) - F(x)}{F'(x)} / \sum_x 1 \quad (2.18)$$

Giriş görüntüsü lineer bir fonksiyon olduğunda, (2.16) yaklaşıklığı ile verilen lineer yaklaşım doğru sonuçlar vermektedir. Tam tersine giriş görüntüsünün ikinci dereceden türevinin $|F''(x)|$ büyük olduğu yerlerde kötü sonuç verdiği kabulü yapılırsa (2.18) yaklaşıklığı ile elde edilen ortalama geliştirilir. (2.18) yaklaşıklığındaki ortalamaya, $|F''(x)|$ tahmininin her noktaya olan etkisi ters orantılı olarak ilave edilir.

$|F''(x)|$ tahmini (2.19) yaklaşıklığı ile gösterilmektedir:

$$F''(x) \approx \frac{G'(x) - F'(x)}{h} \quad (2.19)$$

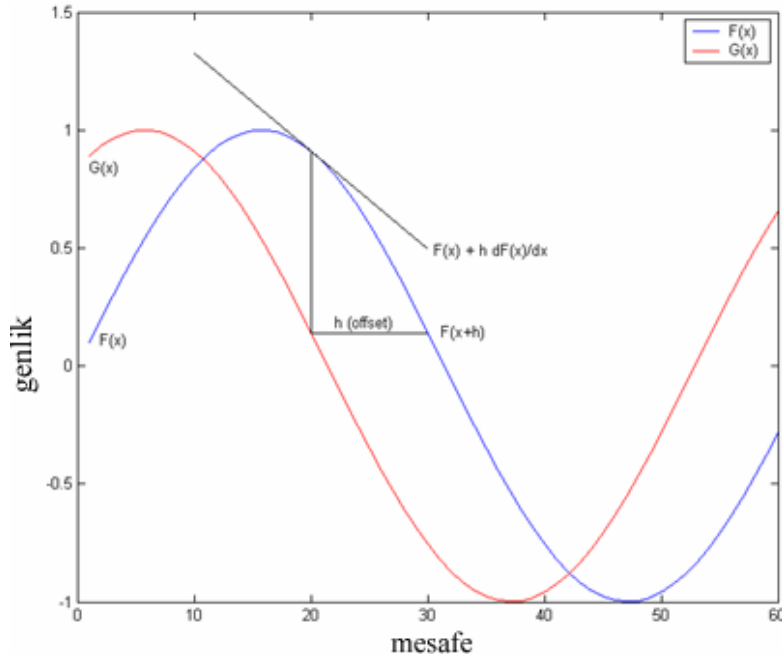
(2.19) yaklaşıklığından elde edilen $1/h$ 'ın sabit faktörü, (2.18) yaklaşıklığında ağırlık fonksiyonu olarak kullanılmaktadır:

$$w(x) = \frac{1}{G'(x) - F'(x)} \quad (2.20)$$

Sonuç olarak $|F''(x)|$ yaklaşıklığı ile verilen ağırlık fonksiyonu bu bölümde bahsedilen sezgisel yaklaşımı karşılamaktadır. Örneğin, Şekil 2.9 'da gösterilen iki eğrinin kesiştiği noktada (2.17) yaklaşıklığı ile sağlanan h değeri "0" a eşittir ve bu durum hız tahmininde istenmeyen bir durumdur. Bununla birlikte, eğrilerin kesişim noktasında $F'(x)$ ve $G'(x)$ arasındaki fark büyük olduğu için ortalamada bu tahmine karşılık gelen ağırlık küçük olacaktır.

(2.20) 'de elde edilen ağırlık faktörünün h hesaplamasına etkisi, (2.21) yaklaşıklığı ile tekrar düzenlenmektedir:

$$h \approx \sum_x \frac{w(x)[G(x) - F(x)]}{F'(x)} / \sum_x w(x) \quad (2.21)$$



Şekil 2.10 Eşleştirilen eğrilerin h mesafesine göre değişimleri.

h yaklaşımı elde edildikten sonra görüntü $F(x)$ hareket ettirilir ve bu işlem Newton-Raphson tekrarlama ile tekrar edilir. İdeal durumlarda sağlanan h tahmin dizileri, en iyi h değeri ile kesişecektir. h tahmininde kullanılan tekrarlama aşağıda (2.22) denklemi ile verilmektedir:

$$h_0 = 0, \quad h_{k+1} = h_k + \sum_x \frac{w(x)[G(x) - F(x + h_k)]}{F'(x + h_k)} / \sum_x w(x) \quad (2.22)$$

burada k tekrarlama sayısını ifade etmektedir.

2.3.2.2 Alternatif Yaklaşım

İki boyutlu görüntü dizilerinde önceki bölümde anlatılan h tahmini için kullanılan tekrarlama ile istenilen sonuçlar elde edilemez. Bunun nedeni, iki boyutlu görüntülerin lineer yaklaşımı ve tek boyutlu işaretin lineer yaklaşımı farklı biçimlerde meydana gelmesidir. Örneğin, (2.17) yaklaşıklığında $F'(x)$ 'in "0" olduğu durumlarda eğim seviyesi belirlenemeyecektir. Bu problemler (2.16) yaklaşıklığının lineer yaklaşımı ile düzeltilebilir:

$$F(x + h) \approx F(x) + hF'(x) \quad (2.23)$$

Hareket değişim mesafesi h 'a bağlı olarak hatayı minimize etmek için hatanın h bileşenine göre türevi alınır:

$$h \approx \frac{\sum_x F'(x)[G(x) - F(x)]}{\sum_x F'(x)^2} \quad (2.24)$$

ve

$$\begin{aligned} 0 &= \frac{dE}{dh} \\ &\approx \frac{d}{dh} \sum_x [F(x) + hF'(x) - G(x)]^2 \\ &= \sum_x 2F'(x)[F(x) + hF'(x) - G(x)] \end{aligned} \quad (2.25)$$

(2.25) yaklaşıklığı ile elde edilen h bileşeni aslında (2.21)'de bahsedilen h yaklaşıklığı ile aynıdır; sadece burada belirlenen ağırlık fonksiyonu $w(x)=F'(x)^2$ 'dir. Özetle bu bölümde elde edilen lineer yaklaşım yardımıyla iki veya daha çok boyutlarda görüntü hareketi tespiti yapılmaktadır. Ayrıca bu yaklaşıklık sayesinde bölen paydanın “0” olması probleminden kaçınılmış olunur; sadece $F'(x)$ 'in görüntünün her yerinde sıfır olduğu durumda yaklaşıklık bir sonuç vermeyecektir. (2.17) yaklaşıklığında ise $F'(x)$ 'in herhangi bir yerde “0”a eşit olması durumunda bir sonuç alınamayacaktır.

$$\begin{aligned} h_0 &= 0, \\ h_{k+1} &= h_k + \frac{\sum_x w(x)F'(x+h_k)[G(x) - F(x+h_k)]}{\sum_x w(x)F'(x+h_k)^2} \end{aligned} \quad (2.26)$$

burada $w(x)$ ağırlık faktörü (2.17) yaklaşıklığında verildiği şekliyle kullanılmaktadır.

2.3.2.3 Performans

Bu bölümde h_k 's dizilerinin hangi koşullar altında ve ne kadar hızlı gerçek h değerinde birleştiği değerlendirilmektedir. Örnek olarak aşağıda (2.27) ile belirtilen durum ele alınmaktadır:

$$\begin{aligned} F(x) &= \sin(x) \\ G(x) &= F(x+h) = \sin(x+h) \end{aligned} \quad (2.27)$$

Her iki kaydetme algoritması da doğru h değerine $|h| < \pi$ 'den önce ulaşacaktır. Yanlış başlangıç kaydetmeleri için en kötü durumda doğru h değerine ulaşma süresi yarım dalga boyu kadar büyük olacaktır. Görüntü yumuşatılarak yüksek uzaysal frekanslar batırılarak doğru h değeri ile kesişme süresi geliştirilir. Örneğin, yumuşatma uygulanarak görüntüdeki her piksel komşu piksellerdeki ağırlıklı ortalamalar ile değiştirilir. Fakat yumuşatmanın kötü bir tarafı ise küçük detayları söndürmesidir ve bu nedenle iki görüntü çerçevesi arasında yapılan eşleştirme daha az doğru olacaktır. Ayrıca eşleştirilmeye çalışılan nesnenin boyutu yumuşatma penceresinden daha küçük ise nesne tamamen söndürülür ve hiçbir eşleşme mümkün olmaz (daha ayrıntılı bilgi için B. D. Lucas, 1984).

Alçak geçiren filtre ile yumuşatma sonucunda elde edilen görüntü bilgi kaybı olmadan düşük çözünürlükte örneklenebilir. Bu düşünceden hareketle coarse-fine stratejisi seçilerek daha doğru tahminler elde edilmektedir. Coarse-fine yöntemi, yaklaşık bir eşleştirme elde edebilmek için görüntünün daha düşük çözünürlüklü yumuşatılmış versiyonlarını kullanmaktadır.

Yumuşatma etkisi doğru h değeri ile kesişme aralığını genişletirken, ağırlık fonksiyonu yaklaşımın doğruluğunu arttırmak için çalışmaktadır ve bu nedenle kesişme süresini azalmaktadır. Ağırlık fonksiyonu kullanılmadığı durumda, diğer bir anlamda $w(x)=1$ için $f(x)=\sin(x)$ giriş işaretine sahip (2.26) denkleminin ilk tekrarında hesaplanan h_1 farkı, mesafe yarım dalga boyuna ulaşır ulaşmaz sıfıra düşer. Bununla birlikte, (2.20) denklemindeki $w(x)$ ağırlığına sahip fark hesabının doğruluğu daha yüksektir ve h farkı, yarım dalga boyuna çok yakın sıfıra düşer. Bu nedenle (2.20) denklemindeki gibi bir $w(x)$ ağırlık faktörüne sahip kesişimler büyük mesafeler için daha hızlı sonuç verecektir.

2.3.2.4 Uygulama

(2.26) denkleminin uygulanabilmesi için görüntü üzerinde R ilgilenilen alan boyunca $F'G, F'F$ ve $(F')^2$ değerlerinin ağırlıklı ortalamaları hesaplanması zorunludur. $F'(x)$ değerinin tam olarak hesaplanması mümkün değildir. Fakat Lucas ve Kanade algoritmasının uygulanabilirliği için $F'(x)$ aşağıdaki yaklaşıklık kullanılarak tahmin edilmektedir:

$$F'(x) = \frac{F(x + \nabla x) - F(x)}{\nabla x} \quad (2.28)$$

ve benzer şekilde $G'(x)$ 'i de bu şekilde hesaplanır. ∇x için küçük bir değer seçilir (yaklaşık

1 piksel). Birinci türevlerin hesaplaması için bazı geliştirilmiş yöntemler bulunmaktadır fakat genelde bu tür yöntemler ilk yumuşatma fonksiyonuna eşittir.

2.3.2.5 Çok Boyutlu Gerçekleme

(2.26) denklemi ile belirtilen tek boyutlu algoritma iki ve ikiden daha fazla boyutlar için de gerçekleştirilebilir. Genel anlamda amaç hata hesabı L_2 normunu minimize etmektir:

$$E = \sum_{x \in R} [F(x+h) - G(x)]^2 \quad (2.29)$$

burada x ve h , n-boyutlu satır vektörleridir. (2.23) yaklaşıklığına benzer şekilde aşağıdaki lineer yaklaşım yapılır:

$$F(x+h) \approx F(x) + h \frac{d}{dx} F(x) \quad (2.30)$$

$\frac{d}{dx}$, burada x 'e bağlı gradyan operatörüdür. $\frac{d}{dx}$ 'in kolon vektörü gösterimi (2.31)'de gösterilmiştir:

$$\frac{d}{dx} = \left[\frac{d}{dx_1} \frac{d}{dx_2} \dots \frac{d}{dx_n} \right]^T \quad (2.31)$$

Bu yaklaşım kullanılarak, hata hesabı L_2 normunu minimize etmek için, aşağıdaki değerleri atanmaktadır:

$$h \approx \left[\sum_x \left(\frac{dF}{dx} \right)^T [G(x) - F(x)] \right] \left[\sum_x \left(\frac{dF}{dx} \right)^T \left(\frac{dF}{dx} \right) \right]^{-1} \quad (2.32)$$

ve

$$\begin{aligned} 0 &= \frac{dE}{dh} \\ &\approx \frac{d}{dh} \sum_x [F(x) + h \frac{dF}{dx}(x) - G(x)]^2 \\ &= \sum_x 2 \frac{dF}{dx}(x) [F(x) + h \frac{dF}{dx}(x) - G(x)] \end{aligned} \quad (2.33)$$

Tek boyutlu durum hesaplamaları ilgili olarak önceki bölümlerde bahsedilen tekraralama, ağırlıklandırma, yumuşatma ve coarse-fine teknikleri, n-boyutlu durumlara da aynı şekilde uygulanmaktadır. İki boyutlu durumda h tahmini için beş çarpımın

$((G-F)F_x, (G-F)F_y, F_x^2, F_y^2$ ve $F_x F_y$) ağırlıklı ortalamasının görüntü üzerinde R ilgilenilen alan boyunca toplanması zorunludur.

2.3.3 Minimize Etme

Lucas-Kanade fark yönteminde olduğu gibi bu tür farksal yöntemler, t ve $t + \delta t$ anlarında alınan iki görüntü çerçevesi arasındaki hareketi her piksel konumunda hesaplamaya çalışmaktadır. Bu yöntemler görüntü işaretinin yerel Taylor serileri yaklaşımına dayandığı için farksal olarak adlandırılırlar; diğer bir anlamda uzaysal ve zamansal koordinatlara göre kısmi türevleri kullanırlar (B. D. Lucas, 1984). $I(x,y,t)$ parlaklığına sahip (x,y,t) konumundaki bir piksel, iki çerçeve arasında δx , δy ve δt ile hareket ederse, aşağıdaki “görüntü kısıtlama denklemi” yazılabilir:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (2.34)$$

Hareketin küçük olduğu varsayılırsa, $I(x,y,t)$ 'deki görüntü bileşeni Taylor serileri ile aşağıdaki denklemle genişletilebilir:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t + YDT \quad (2.35)$$

burada YDT, yüksek dereceden terimler anlamına gelmektedir; ve bu terimler ihmal edilebilecek kadar küçüktür. Bu eşitliklerin devamında aşağıdaki sonuçlar elde edilmektedir:

$$\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0 \quad (2.36)$$

veya

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} \frac{dt}{dt} = 0$$

bunun sonucu olarak hız bileşenlerini içeren denklem elde edilir:

$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0 \quad (2.37)$$

burada v_x ve v_y ; $I(x,y,t)$ 'nin optik akışı veya hızın x ve y bileşenleridir ve $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ ve $\frac{\partial I}{\partial t}$;

(x,y,t) pikselindeki görüntünün ilişkin yönlerdeki türevleridir. (2.37) denklemindeki türevler

yerine I_x , I_y ve I_t bileşenleri yazılabilir:

$$I_x v_x + I_y v_y = -I_t \quad (2.38)$$

veya

$$\nabla I \cdot \vec{v} + I_t = 0 \quad (2.39)$$

Bu eşitlik iki bilinmeyenlidir ve bu şekilde çözülemez. Bu durum optik akış algoritmalarının açıklık problemi olarak bilinmektedir. Optik akışı bulabilmek için bazı ek bileşenler veren başka bir eşitlik seti gerekmektedir. Lucas ve Kanade tarafından elde edilen çözüm, yerel olarak sabit bir akış kabulü yapan yinelemesiz bir yöntemdir.

$m \times m$ boyutundaki küçük bir pencere içerisinde ($m > 1$), merkezi (x, y) pikselinde yer alan akışın (v_x, v_y) sabit olduğu kabulü ile bir denklem seti bulunabilir:

$$\begin{aligned} I_{x1} v_x + I_{y1} v_y &= -I_{t1} \\ I_{x2} v_x + I_{y2} v_y &= -I_{t2} \\ &\vdots \\ &\vdots \\ &\vdots \\ I_{xn} v_x + I_{yn} v_y &= -I_{tn} \end{aligned} \quad (2.40)$$

İki bilinmeyen için iki denklemden daha fazla denklem bulunduğundan sistem aşırı-kararlıdır.

Bu nedenle aşağıdaki ifade yazılabilir:

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ -I_{t2} \\ \vdots \\ \vdots \\ -I_{tn} \end{bmatrix} \quad (2.41)$$

veya

$$A\vec{v} = -b$$

Aşırı-kararlı sistemin denklemlerini çözmek için, Lucas ve Kanade optik akış tahmininde “en küçük kareler” yöntemi kullanılır:

$$A^T A \vec{v} = A^T (-b) \text{ veya}$$

$$\vec{v} = (A^T A)^{-1} A^T (-b)$$

veya

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \sum I_{xi}^2 & \sum I_{xi} I_{yi} \\ \sum I_{xi} I_{yi} & \sum I_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_{xi} I_{ti} \\ -\sum I_{yi} I_{ti} \end{bmatrix} \quad (2.42)$$

bu denklemde toplamalar $i=1$ 'den n 'e kadar çalıştırılmaktadır.

Bunun anlamı, tüm üç boyut içerisinde görüntü türevleri hesaplanarak optik akış bulunabilir. $i, j \in [1, \dots, m]$ bileşenlerine sahip $W(i, j)$ ağırlık fonksiyonu eklenerek pencere merkezine daha fazla önem verilebilir. Bu amaçla Gauss fonksiyonları tercih edilir. Diğer fonksiyonlar veya ağırlık şemaları da kullanılmaktadır. Ayrıca yerel çevrimleri hesaplamada, görüntü bozulmalarını benzeştirmek için de optik akış modeli genişletilebilir.

Görüntü üzerinde Ω penceresi küçültülerek her küçük uzaysal komşudaki $\vec{v}$ için sabit bir modele yerel birinci dereceden bileşenlerin ağırlıklı en küçük kareleri uygulanmaktadır:

$$E(v_x, v_y) = \sum_{x, y \in \Omega} W^2(x, y) [\nabla I(x, y, t) \cdot \vec{v} + I_t(x, y, t)]^2 \quad (2.43)$$

Yukarıdaki denklemin çözümü şu şekilde verilir:

$$\vec{v} = [A^T W^2 A]^{-1} A^T W^2 \vec{b} \quad (2.44)$$

N piksel için ($n \times n$ komşuluk, $N=n^2$), t anında $(x_i, y_i) \in \Omega$ kabulü ile yukarıdaki denklemin çözümünde gerekli bileşenler aşağıdaki eşitliklerle hesaplanmaktadır:

$$\begin{aligned} A &= [\nabla I(x_1, y_1), \dots, \nabla I(x_N, y_N)], \\ W &= \text{diag}[W(x_1, y_1), \dots, W(x_N, y_N)], \\ b &= -(I_t(x_1, y_1), \dots, I_t(x_N, y_N)). \end{aligned} \quad (2.45)$$

$A^T W^2 A$ tek olmayan matris ise denklemi kapalı formda çözülebilir. $A^T W^2 A$, 2×2 'lik matristir:

$$A^T W^2 A = \begin{bmatrix} \sum W^2(x, y) I_x^2(x, y) & \sum W^2(x, y) I_x(x, y) I_y(x, y) \\ \sum W^2(x, y) I_y(x, y) I_x(x, y) & \sum W^2(x, y) I_y^2(x, y) \end{bmatrix}, \quad (2.46)$$

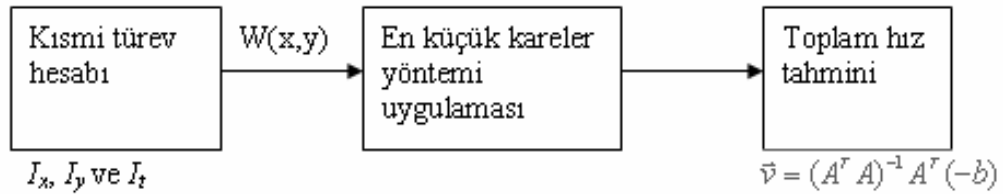
burada tüm toplamalar Ω komşuluğunda (x, y) pikseli boyunca alınmaktadır.

Stereo eşleştirmede olduğu gibi görüntü kaydetme uygulandığında, Lucas-Kanade metodu genellikle *coarse-fine* tekrarlama davranışına taşınır, bu tür bir yolda ilk olarak uzaysal türevler ölçek-uzayında (veya piramit) *coarse* ölçeğinde hesaplanır, görüntülerden biri hesaplanan bozulma ile çevrilir, ve daha sonra tekrarlama güncellemeleri *finer* ölçeklerde başarılı bir şekilde hesaplanır.

Lucas-Kanade algoritması ve diğer yerel optik akış algoritmalarının karakteristiklerinden biri de çok yüksek akış vektörü yoğunluğu sağlamamasıdır; bir diğer değişle akış bilgisi, hareket sınırları boyunca hızlıca kaybolur ve büyük homojen alanların iç bölümleri küçük hareketleri gösterir. Bunun avantajı, gürültü varlığında karşılaştırılmalı sağlamalıdır.

2.3.4 Temel Lucas Kanade Metodu MATLAB Programı Uygulaması

Lucas ve Kanade algoritmasının blok diyagramı Şekil 2.11 'de gösterilmiştir. Blok diyagramda da gösterildiği üzere öncelikle görüntü dizileri yumuşatılır ve yüksek frekanslı bileşenler atılır. Daha sonra belirlenen pencere büyüklüğüne bağlı olarak gradyan tahmini yapılır ve 2×2 'lik $A^T W^2 A$ matrisi elde edilir. Pencerenin tüm görüntü üzerinde gezdirilmesi sonucunda elde edilen hız değerleri çıkışa verilir.



Şekil 2.11 Lucas-Kanade metodunun blok diyagramı.

2.3.4.1 Test Sonuçları

Lucas ve Kanade optik akış algoritması MATLAB uygulaması ile gerçekleştirilmiştir. Kullanılan pencere büyüklüğü 5, 10 ve 15 değerleri verilerek istenen sonuca ulaşılmıştır. Görüntü boyunca hareket eden nesnelerin büyüklükleri ile pencere alanı seçimi arasında doğru orantı bulunmaktadır. Piksel bazlı hız tahmini yerine bloklar halinde hız tahmini metodu uygulandığı için sonucun gürültüden etkilenme oranı da azalacaktır. Fakat bunun sonucu olarak küçük nesnelerin hareketi algılanamaz.

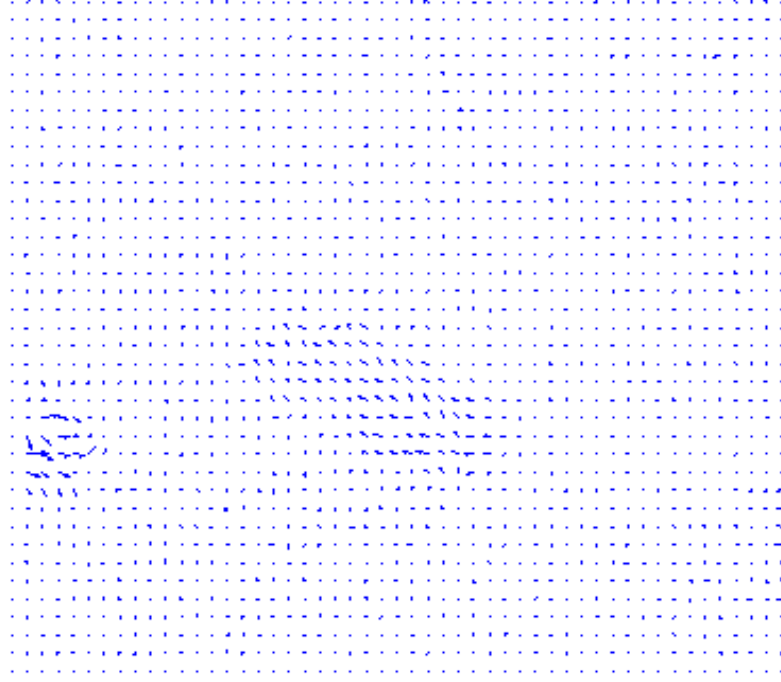


(a)

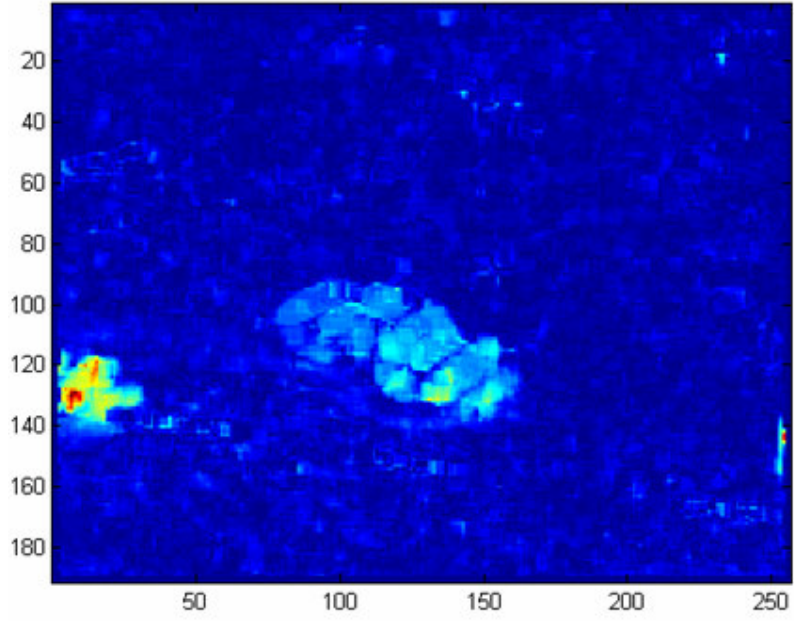


(b)

Şekil 2.12 (a), (b) Orijinal görüntü dizileri.

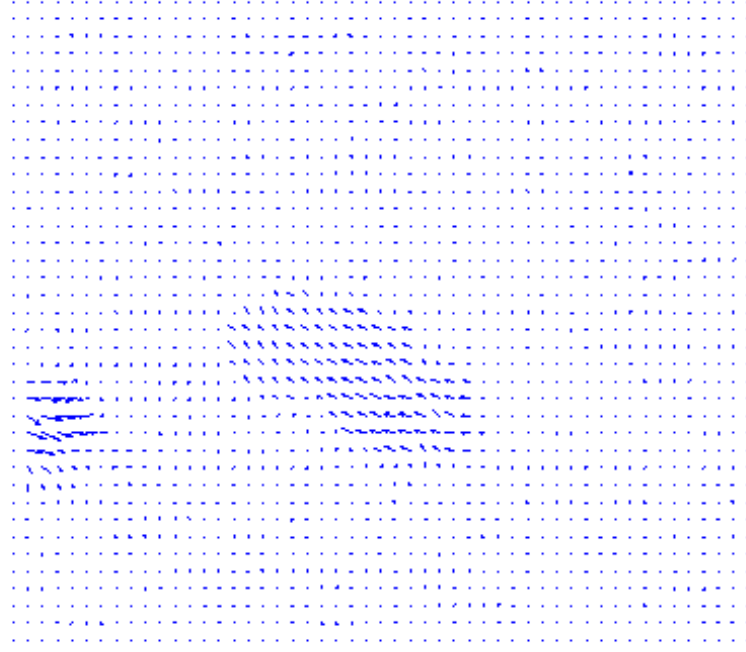


(a)

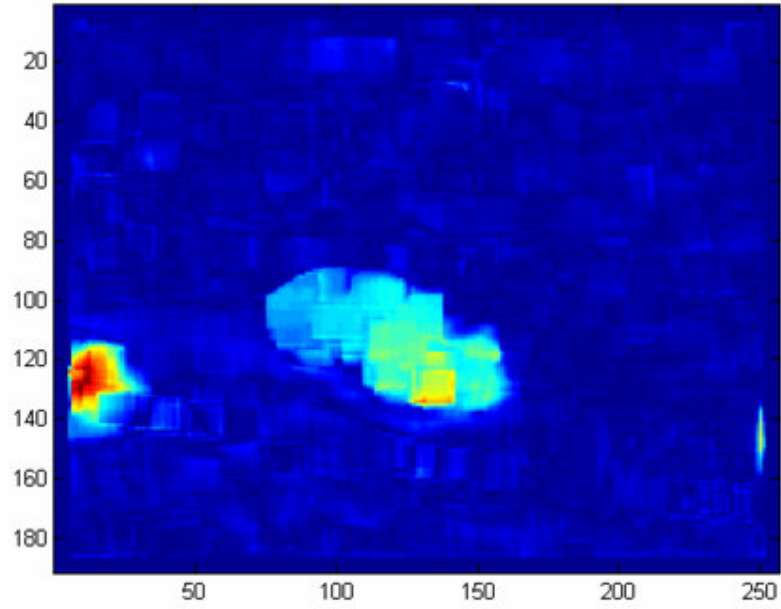


(b)

Şekil 2.13 5x5 pencere için (a) Optik akış alanı, (b) Enerji alanı

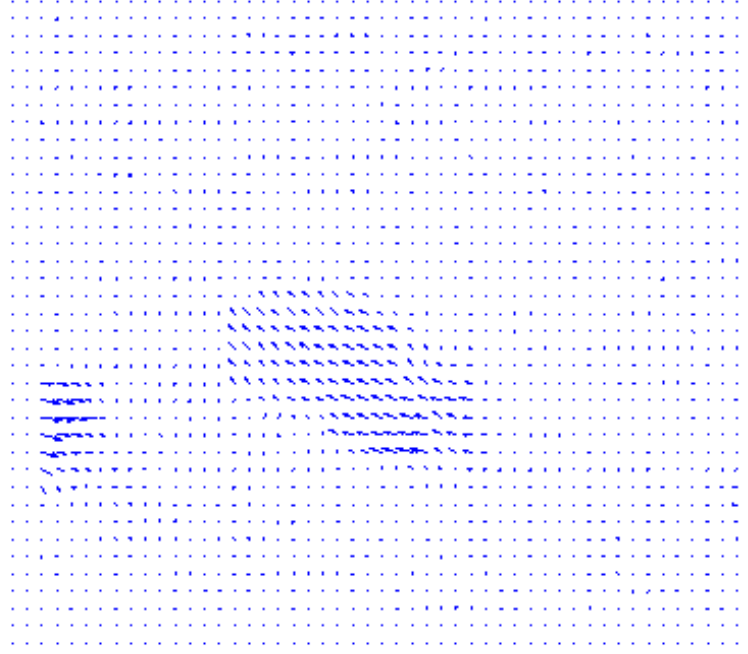


(a)

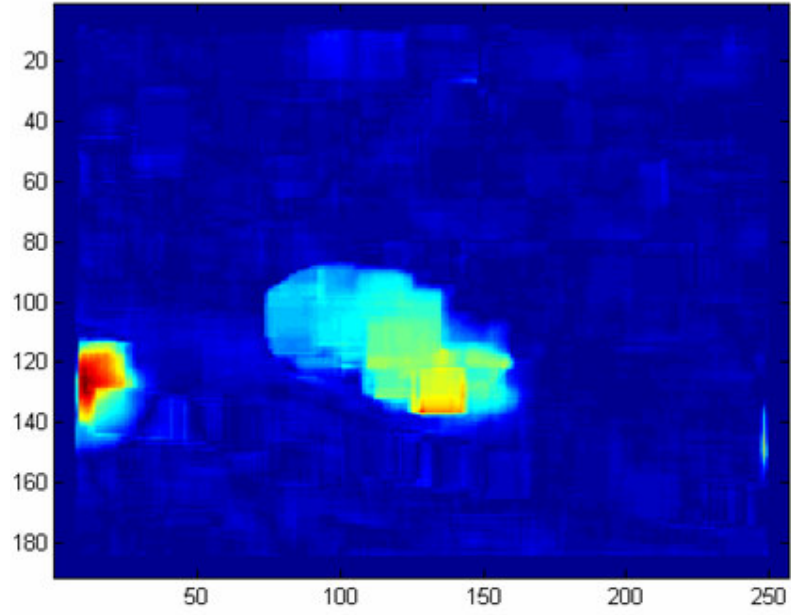


(b)

Şekil 2.14 10x10 pencere için (a) Optik akış alanı, (b) Enerji alanı.



(a)



(b)

Şekil 2.15 15x15 pencere için (a) Optik akış alanı, (b) Enerji alanı.

2.3.5 Lucas Kanade Yönteminin Gauss Piramidi Uygulaması

2.3.5.1 Gauss Ölçek-Uzayı (Piramidi)

Fizik ve biyolojik görmenin de katkılarıyla bilgisayarla görme, görüntü işleme ve işaret işleme tarafından geliştirilen ölçek-uzayı (veya piramit) teorisi; çok-ölçekli işaret gösterim için bir iskelettir. Ölçek-uzayı teorisi, yumuşatılmış görüntülerin bir-parametreliliğine ailesine orijinal görüntüyü gömme ile farklı ölçeklerdeki görüntü yapılarını elde eden biçimsel bir teoridir; ölçek-uzayı seviyeleri, yumuşatma (alçak geçiren filtre) maskesinin boyutu ile parametreleşir. Bu ailedeki t parametresi, ölçek parametresine karşılık gelmektedir; yaklaşık olarak $\sqrt{t}$ 'den daha küçük uzaysal boyutun görüntü yapıları, t ölçeğindeki ölçek-uzayı seviyesi boyunca büyük ölçüde yumuşatılmaktadır.

Ölçek-uzayı iskeleti ayrıca Gauss türev operatörlerine dayanan eski görsel çalışmalar için bir teoriyi de içerir; bu teori, otomatik olarak görsel bilgiyi işleyen bilgisayarlı sistemler için eski görsel çalışmaların büyük bir sınıfını anlatmak için bir temel olarak kullanılabilir. Gerçek dünya nesneleri farklı boyutlarda olabilir ve ek olarak nesne ve kamera arasındaki mesafe bilinmeyebilir ve koşullara bağlı olarak değişebilir; bu gerçeklere bağlı olarak görüntü verisinde medyana gelebilecek boyut değişimleri ile baş edebilmek için, görsel çalışmaların ölçek değişmezliğinden yapılmasına da izin verir (T. Lindeberg, 1994).

2.3.5.2 Ölçek-Uzayı Tanımı

Ölçek-uzayı gösterimi, değişkenlerin keyfi sayılarının işaretlerine uygulanır. Literatürdeki en yaygın örnek, iki boyutlu görüntülere uygulanmaktadır. Verilen görüntü $f(x, y)$ için, lineer (Gauss) ölçek-uzayı gösterimi, türetilmiş işaretleri ailesi $L(x, y; t)$, ölçek-uzayı seviyeleridir; bu da Gauss maskesi ile $f(x, y)$ 'nin konvolüsyonu tarafından tanımlanır.

$$g(x, y; t) = \frac{1}{2\pi t} e^{-\frac{(x^2 + y^2)}{2t}} \quad (2.47)$$

şu şekilde gösterilir;

$$L(x, y; t) = g(x, y; t) * f(x, y) \quad (2.48)$$

burada g argümanındaki noktalı virgül konvolüsyonun x ve y değerleri üzerinde gerçekleştirildiğini gösterir; noktalı virgülden sonraki ölçek parametresi t , sadece ölçek-uzayı

seviyesini tanımladığını gösterir. L 'nin bu gösterimi, $t \geq 0$ ölçeklerinin devamlı olması için çalışır fakat tipik olarak sadece ölçek-uzayı seviyesinin sınırlı bir ayrık seti gerçekte düşünülür.

t , Gauss filtresinin bir değişkenidir ve $t = 0$ için elde edilen filtre g , dürtü fonksiyonu olur; şöyle ki $L(x,y,0) = f(x,y)$, ölçek parametresi 0'a karşılık gelen ölçek-uzayı seviyesi, görüntünün f kendisidir. t arttıkça, L ; daha ve daha büyük bir filtre ile içerdigi daha ve daha fazla detayı silerek yumuşatılan f 'in sonucu olur. Filtrenin standart sapması $\sqrt{t}$ 'dir.



Şekil 2.16 (a) Ölçek-uzayı seviyesi $L(x,y,0)$, f görüntüsüne karşılık gelir.



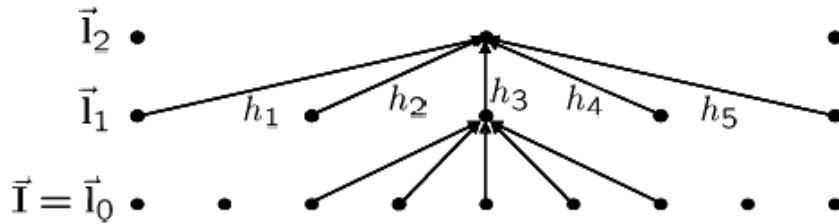
Şekil 2.16 (b) Ölçek-uzayı seviyesi $L(x,y,1)$.



Şekil 2.16 (c) Ölçek-uzayı seviyesi $L(x,y,4)$.

Daha önceki bölümlerde bahsedilen Lucas ve Kanade optik akış algoritması Gauss Piramidi metodu da ilave edilerek geliştirilmektedir. Coarse-fine yöntemi olarak da adlandırılan bu yöntem, görüntüyü gerçek boyutundan belli bir çözünürlüğe düşürerek seviyeler ayırır. En düşük çözünürlükte Lucas ve Kanade algoritması uygulanarak elde edilen hız tahmini değerleri sonraki seviyeye giriş olarak verilir. Böylece daha iyi bir hız tahmini sağlanmaktadır.

Alçak geçiren filtreden geçirilerek aşağı-örneklenen görüntü dizileri $[\vec{I}_0, \vec{I}_1, \dots, \vec{I}_N]$ matrisi ile gösterilmektedir. Genellikle aşağı-örnekleme işlemi tek boyutlu ayrıştırılabilir kernel $h=[h_1, h_2, h_3, h_4, h_5]$ ile yapılır ve her yönde aşağı-örnekleme faktörü 2'dir (Adelson, Anderson, Bergen, Burt, and Ogden, 1984).



Şekil 2.17 Aşağı örnekleme gösterimi.

Bir seviyeden diğerine matris gösterimindeki adresleme, (2.49) matris formu ile gösterilmektedir:

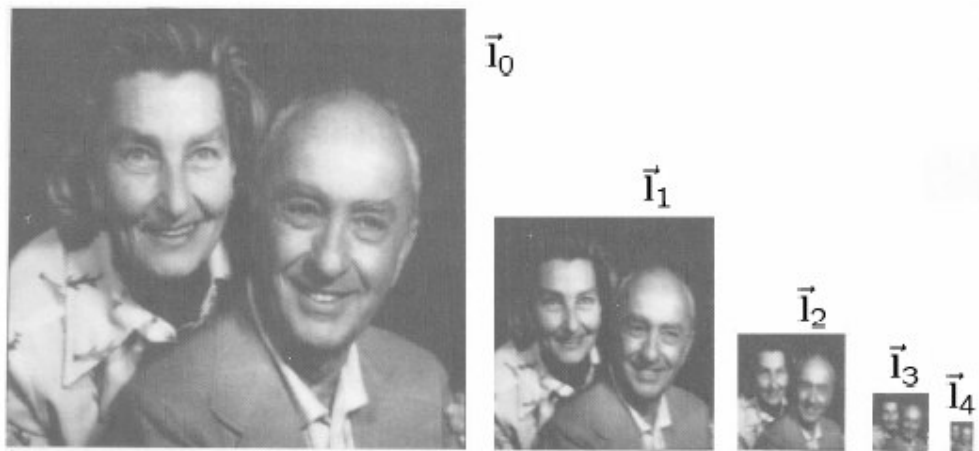
$$\vec{I}_{k+1} = R \vec{I}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 \\ \cdot & & & \cdot & \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \\ \cdot \\ -h- \\ -h- \\ -h- \\ \cdot \\ \cdot \end{bmatrix} \vec{I}_k \quad (2.49)$$

aşağı-örnekleme konvolüsyon

İki terimli katsayıların impuls cevabı için kullanılan tipik ağırlıklar:

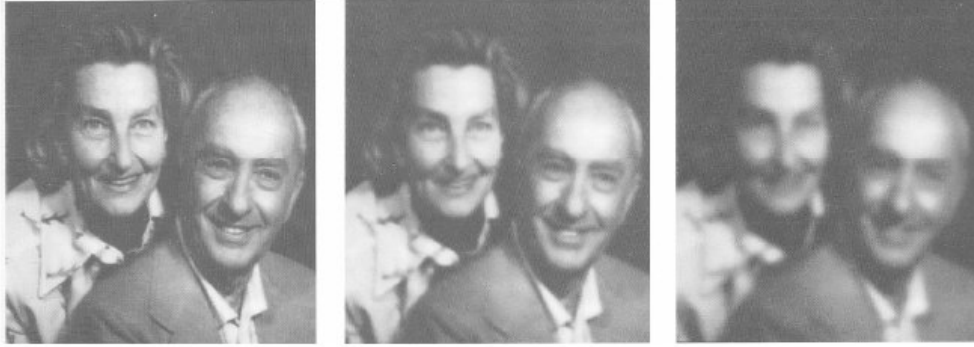
$$h = \frac{1}{16}[1, 4, 6, 4, 1] \quad (2.50)$$

Orijinal görüntü ($\vec{I}_0$) ve beş seviyeli gauss piramidi örneği Şekil 2.18 'de gösterilmektedir:



Şekil 2.18 Örnek görüntünün 4 seviyeli ($L(x,y,4)$) Gauss Piramidi uygulaması.

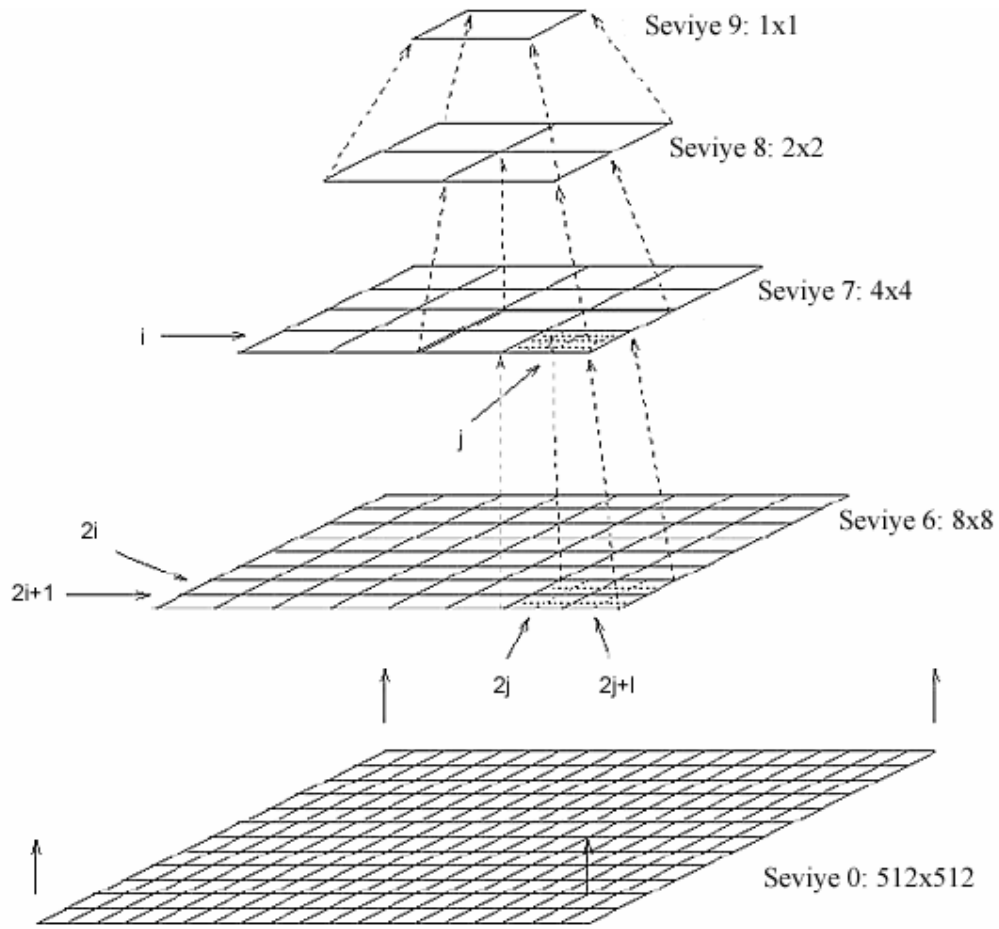
Şekil 2.19 ‘da ilk üç seviye orijinal görüntü ile aynı büyüklükte olacak şekilde ölçeklendirilmiştir:



Şekil 2.19 Örnek görüntünün Gauss Piramidi uygulamasının 2 seviyesi için aynı boyuttaki gösterimi.

Gauss piramidin özellikleri aşağıdaki şekilde özetlenebilir:

- Birden çok ölçekli eniyileme sağlama,
- İlgilenilen alanı izole etme,
- Gürültü etkisini azaltma.



Şekil 2.20 512x512’lik görüntünün 9 seviyeli ($L(x,y,9)$) Gauss Piramidi gösterimi.

2.3.5.3 Problem Tanımı

I ve J görüntülerinin 2 boyutlu gri-ölçekli görüntüler olduğu kabul edilmektedir. İki nicelik $I = I(x, y)$ ve $J = J(x, y)$, $x = [x \ y]^T$ konumundaki iki görüntünün gri-ölçekli değerleridir; burada x ve y pikselleri genel bir görüntü noktası x ‘in koordinatlarıdır. Görüntü I , genellikle ilk görüntü olarak tanımlanır ve görüntü J , ikinci görüntüdür. Pratikte I ve J görüntüleri ayrık fonksiyondur (veya dizidir), ve sol en üst köşe piksel koordinatı $[0 \ 0]^T$ ‘dır. n_x ve n_y , iki görüntünün genişlik ve yüksekliğidir. En alt sağ piksel koordinat vektörü; $[n_x - 1 \ n_y - 1]^T$.

İlk görüntü I üzerindeki bir görüntü noktası $u = [u_x \ u_y]^T$ olarak düşünülmektedir. Buradaki

amaç; $I(u)$ ve $J(v)$ 'nin benzer olması gibi ikinci görüntü J üzerindeki konumunu $v = u + d = [u_x + d_x \ u_y + d_y]^T$ bulmaktır. Vektör $d = [d_x \ d_y]^T$, x konumundaki görüntü hızıdır; ayrıca x noktasındaki optik akış olarak da bilinir. Açıklık problemi yüzünden, 2 boyutlu komşuluk anlayışında benzerlik notasyonunu belirlemek önemlidir. w_x ve w_y iki tam sayıdır. Aşağıdaki şekilde tanımlanan artık fonksiyonunu $\mathcal{E}$ minimize eden vektör olan görüntü hızı d tanımlanmaktadır:

$$\mathcal{E}(d) = \mathcal{E}(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (I(x, y) - J(x + d_x, y + d_y))^2 \quad (2.51)$$

Benzerlik fonksiyonu, $((2w_x + 1)x(2w_y + 1))$ boyutundaki görüntü komşuluğunda hesaplanmaktadır (J.Y. Bouguet, 1999).

2.3.5.4 Hız Tahmini Algoritması

Hız tahmininde iki anahtar bileşen, doğruluk ve sağlamlıktır. Doğruluk bileşeni, hız tahminine ilişkin yerel alt-piksel doğruluğuna bağlıdır. Sezgisel olarak, küçük entegrasyon penceresi, görüntünün içerdiği detayları “yumuşatmayacak” şekilde tercih edilir (başka bir deyişle w_x ve w_y 'nin küçük değerleri seçilir). Bu özellik; iki yamanın çok farklı hızlarda hareket ettiği görüntüdeki kapatılan bölgelerde gereklidir.

Sağlamlık bileşeni ise, şiddet değişimleri, görüntü hareketi boyutu gibi bileşenlere bağlı olarak hız tahmini duyarlılığı ile ilişkilidir. Özellikle, büyük hareketlerle baş edebilmek için sezgisel olarak büyük entegrasyon penceresi seçilmesi tercih edilir. Gerçekte, sadece denklem 1 ele alınırsa; (bazı öncelikli eşleştirme bilgisi mevcut olmadığı sürece) $d_x \leq w_x$ ve $d_y \leq w_y$ alınması daha uygundur. Bu nedenle, entegrasyon penceresinin boyutunu seçerken, yerel doğruluk ve sağlamlık arasında doğal bir ilişki vardır. Bu probleme çözüm olarak klasik Lucas-Kanade algoritmasının piramit uygulaması önerilmektedir. Lucas-Kanade optik akış hesabının tekrarlamalı uygulaması, yeterli yerel hız tahmini doğruluğu sağlamaktadır.

2.3.5.5 Görüntü Piramidi Gösterimi

$n_x \times n_y$ boyutundaki genel görüntü I 'nın piramit gösterimi tanımlanmaktadır. $I^0 = I$, “sıfırıncı” seviyedeki görüntüdür. Bu görüntü aslında en yüksek çözünürlükteki görüntüdür (ham görüntü). Bu seviyedeki görüntü genişliği ve yüksekliği, $n_x = n_x^0$ ve $n_y = n_y^0$ olarak

tanımlanır. Piramit gösterimi bu şekilde tekrarlamalı bir biçimde geliştirilir: I^1 , I^0 ‘dan hesaplanır ve sonrasında I^2 , I^1 ‘den hesaplanır ve bu şekilde devam eder. $L = 1, 2, \dots$, genel piramit seviyeleridir ve I^{L-1} , $L-1$ seviyesindeki görüntüdür. n_x^{L-1} ve n_y^{L-1} , I^{L-1} görüntüsünün genişlik ve yüksekliğidir. Görüntü I^{L-1} , aşağıdaki şekilde tanımlanabilir:

$$\begin{aligned}
 I^L(x, y) = & \frac{1}{4} I^{L-1}(2x, 2y) + \\
 & \frac{1}{8} (I^{L-1}(2x-1, 2y) + I^{L-1}(2x+1, 2y) + I^{L-1}(2x, 2y-1) + I^{L-1}(2x, 2y+1)) + \\
 & \frac{1}{16} (I^{L-1}(2x-1, 2y-1) + I^{L-1}(2x+1, 2y+1) + I^{L-1}(2x-1, 2y+1) + I^{L-1}(2x+1, 2y-1))
 \end{aligned} \quad (2.52)$$

Notasyonda basitlik için, görüntünün I^{L-1} ($0 \leq x \leq n_x^{L-1} - 1$ ve $0 \leq y \leq n_y^{L-1} - 1$ için) bir piksel çevresindeki test görüntü değerleri tanımlanırsa:

$$\begin{aligned}
 I^{L-1}(-1, y) & \doteq I^{L-1}(0, y), \\
 I^{L-1}(x, -1) & \doteq I^{L-1}(x, 0), \\
 I^{L-1}(n_x^{L-1}, y) & \doteq I^{L-1}(n_x^{L-1} - 1, y), \\
 I^{L-1}(x, n_y^{L-1}) & \doteq I^{L-1}(x, n_y^{L-1} - 1), \\
 I^{L-1}(n_x^{L-1}, n_y^{L-1}) & \doteq I^{L-1}(n_x^{L-1} - 1, n_y^{L-1} - 1).
 \end{aligned} \quad (2.53)$$

2.53 denklemi, sadece x ve y değerleri için tanımlanır; şöyle ki $0 \leq 2x \leq n_x^{L-1} - 1$ ve $0 \leq 2y \leq n_y^{L-1} - 1$ ‘dır. Bu nedenle, aşağıdaki iki koşulu sağlayan I^L ‘nın genişlik n_x^L ve yükseklik n_y^L değerleri en büyük tam sayılardır:

$$n_x^L \leq \frac{n_x^{L-1} - 1}{2} \quad (2.54)$$

$$n_y^L \leq \frac{n_y^{L-1} - 1}{2} \quad (2.55)$$

İki görüntü I ve J : $\{I^L\}_{L=0, \dots, L_m}$ ve $\{J^L\}_{L=0, \dots, L_m}$ ‘nin piramit gösterimlerini tekrarlamalı olarak elde etmek için (2.53), (2.54) ve (2.55) denklemleri kullanılır. L_m değeri, piramidin yüksekliğidir. L_m ‘nin pratik değerleri 2,3,4 ‘tür. Tipik görüntü boyutları için, seviye 4’ten daha yukarıya gidilmesi mantıklı değildir. Örneğin, 640x480 boyutundaki bir I görüntüsü için, I^1, I^2, I^3 ve I^4 görüntüleri sırasıyla; 320x240, 160x120, 80x60 ve 40x30 boyutlarındadır.

Seviye 4 'ün ilerisine gitmek çoğu durumda pek fazla mantı değildir. Piramit gösteriminin arkasındaki merkezi motivasyonu, (w_x ve w_y entegrasyon pencere boyutlarından daha büyük) büyük piksel motivasyonlarını elde edebilir. Bu nedenle, piramit yüksekliği (L_m) de, görüntüde beklenen maksimum optik akışa göre uygun bir şekilde seçilebilir. Bir sonraki bölümde detaylı olarak tanımlanan hız tahmini çalışması bu kavramın daha iyi anlaşılmasını sağlamaktadır. Final görüş: denklem 2.53; görüntü alt-örneklemeden önce görüntü örtüşmesini önlemek için alçak geçiren filtre ($[1/4 \ 1/2 \ 1/4]_x [1/4 \ 1/2 \ 1/4]_y$) kullanılmasını önerir.

2.3.5.6 Piramit Hız Tahmini

Hız tahminin amacı tekrar hatırlanırsa; görüntü I 'da verilen nokta u için, görüntü J 'de bu noktaya ilişkin konumu $v = u + d$ bulma veya alternatif olarak ilişkin piksel yer değiştirme vektörü d 'yi bulma.

$L = 0, \dots, L_m$ için, piramidal görüntüler I^L üzerindeki u noktasının ilişkin koordinatlarını, $u^L = [u_x^L \ u_y^L]^T$ belirleme. Piramit gösterim denklemlerinin tanımını izleyerek, u^L vektörleri aşağıdaki denklemde hesaplanmaktadır:

$$u^L = \frac{u}{2^L} \quad (2.56)$$

Denklem 2.56 'daki bölme operatörü, her iki koordinata da bağımsızca uygulanır. Özellikle $u^0 = u$ olduğu göz önüne alınmalıdır.

Toplam piramidal hız tahmini algoritması şu şekilde devam eder: ilk olarak optik akış en alt piramit seviyesi L_m 'de hesaplanır. Daha sonra, bu hesaplamanın sonucu ($L_m - 1$ seviyesindeki) piksel değiştirme için başlangıç tahmini olarak üstteki seviye $L_m - 1$ 'ye taşınır. Verilen başlangıç tahmini ile artırılmış optik akış, $L_m - 1$ seviyesinde hesaplanır, ve sonuç $L_m - 2$ seviyesine taşınır ve bu şekilde 0 seviyesine (orijinal görüntü) kadar bu işlem devam eder.

Daha matematiksel olarak $L+1$ ve L seviyeleri arasındaki tekraralama işlemi tanımlanabilir. L seviyesindeki optik akış için başlangıç değeri, $g^L = [g_x^L \ g_y^L]^T$; L_m seviyesinden $L+1$ seviyesine kadar yapılan hesaplamalarda mevcuttur. Daha sonra, L seviyesindeki optik akışı

hesaplamak amacıyla, yeni görüntü eşleştirme hata fonksiyonu $\mathcal{E}^L$ 'nu minimize eden artan piksel yer değiştirme vektörünü, $d^L = [d_x^L \ d_y^L]^T$ bulmak önemlidir:

$$\mathcal{E}^L(d^L) = \mathcal{E}^L(d_x^L, d_y^L) = \sum_{x=u_x^L-w_x}^{u_x^L+w_x} \sum_{y=u_y^L-w_y}^{u_y^L+w_y} (I^L(x, y) - J^L(x + g_x^L + d_x^L, y + g_y^L + d_y^L))^2 \quad (2.57)$$

Bütün L değerleri için entegrasyon penceresi sabit boyuttadır, $(2w_x + 1) \times (2w_y + 1)$. Başlangıç tahmini vektörü g^L , ikinci görüntü J 'de görüntü yamasının ön çevrimi için kullanılır. Bu yolla, artan akış vektörü $d^L = [d_x^L \ d_y^L]^T$, küçüktür ve bu nedenle standart Lucas Kanade adımı ile hesaplamak kolaydır.

Artan optik akışın d^L hesaplama detayları bir sonraki bölümde anlatılmaktadır. Algoritmanın ana döngüsünü kapatmak için bu değer hesaplandığı varsayılmaktadır. Daha sonra, bu hesaplamanın sonucu yeni başlangıç tahmini g^{L-1} 'yi ekleyerek bir sonraki seviye $L-1$ 'ye aktarılır.

$$g^{L-1} = 2(g^L + d^L) \quad (2.58)$$

Sonraki seviye optik akış artan vektörü, d^{L-1} aynı prosedürlerle hesaplanır. Optik akış hesaplaması ile hesaplanan bu vektör, $\mathcal{E}^{L-1}(d^{L-1})$ fonksiyonunu minimize eder. Bu prosedür, en iyi görüntü çözünürlüğüne ulaşana kadar devam eder ($L=0$). Algoritma, L_m seviyesi için başlangıç tahminini sıfıra ayarlayarak başlatılır (piramidin en alt seviyesinde başlangıç tahmini mevcut değildir):

$$g^{L_m} = [0 \ 0]^T \quad (2.59)$$

Böylece en iyi optik akış hesabından sonra, final optik akışı çözümü elde edilir:

$$d = g^0 + d^0 \quad (2.60)$$

Bu çözüm, aşağıdaki genişletilmiş biçimde tanımlanabilir:

$$d = \sum_{L=0}^{L_m} 2^L d^L \quad (2.61)$$

Piramit uygulamasının açık avantajı; büyük bir toplam piksel yer değiştirmesi vektörü d 'yi hesaplarken, artan her d^L optik akış vektörünü çok küçük tutulabilmesidir. Her basit optik

akış hesabı adımı, d_{maks} 'a kadar piksel hareketlerini elde edebilir; piramit uygulamasının elde edebildiği toplam piksel hareketi, $d_{maks\ final} = (2^{L_m+1} - 1)d_{maks}$ olur. Örneğin; piramit derinliği $L_m = 3$ için, maksimum yer değiştirme kazancının 15 olduğu anlamına gelir. Bu özellik, entegrasyon penceresi boyutunu küçük tutarken büyük piksel hareketlerine olanak tanır.

2.3.5.7 Tekrarlamalı Lucas ve Kanade Optik Akış Hesaplaması

Optik akış hesaplamasındaki amaç 2.57 denklemdaki eşleştirme fonksiyonu $\mathcal{E}^L$ 'yi en aza indirmek için piramitteki her L seviyesinde d^L vektörünü bulmaktır. Her L seviyesi için aynı işlem tekrarlanır:

$$\forall (x, y) \in [p_x - w_x - 1, p_x + w_x + 1] \times [p_y - w_y - 1, p_y + w_y + 1] \quad (2.62)$$

$$A(x, y) \doteq I^L(x, y)$$

$$\forall (x, y) \in [p_x - w_x, p_x + w_x] \times [p_y - w_y, p_y + w_y] \quad (2.63)$$

$$B(x, y) \doteq J^L(x + g_x^L, y + g_y^L)$$

$A(x, y)$ ve $B(x, y)$ tanımlamalarının domenlerinin birbirinden biraz farklı olduğunu gözlemlenir. Gerçekten de $A(x, y)$ tanımlamasının pencere boyutu $(2w_x + 1) \times (2w_y + 1)$ olması gerekirken $(2w_x + 3) \times (2w_y + 3)$ olmuştur. Bu fark, merkezi fark operatörünü kullanarak hesaplanan $A(x, y)$ uzaysal türeviyle daha da belirginleşiyor (denk. 2.70 ve 2.71). Daha anlaşılır olması için yer değiştirme vektörü $\vec{v} = [v_x, v_y]^T = d^L$ değiştirilir; tabi ki görüntü pozisyon vektörü $p = [p_x, p_y]^T = u^L$ 'de değiştirilir. Yeni notasyonu bulmak için hedef yer değiştirme vektörünü $\vec{v} = [v_x, v_y]^T$ değiştirmektedir.

$$\mathcal{E}(\vec{v}) = \mathcal{E}(v_x, v_y) = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (A(x, y) - B(x + v_x, y + v_y))^2 \quad (2.64)$$

Standart tekrarlamalı Lucas-Kanade bu iş içinde uygulanabilir. Optimumda, $\mathcal{E}$ 'nin $\vec{v}$ 'ye ilişkin birinci türevi sıfırdır.

$$\left. \frac{\partial \mathcal{E}(\vec{v})}{\partial \vec{v}} \right|_{\vec{v}=\vec{v}_{opt}} = [0 \ 0] \quad (2.65)$$

Türevin genişletilmesinden sonra elde edilen,

$$\frac{\partial \mathcal{E}(\vec{v})}{\partial \vec{v}} = -2 \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (A(x, y) - B(x+v_x, y+v_y)) \cdot \left[\frac{\partial B}{\partial x} \frac{\partial B}{\partial y} \right] \quad (2.66)$$

Bundan sonra $B(x+v_x, y+v_y)$ teriminin yerine noktayla ilgili $\vec{v} = [0 \ 0]^T$ 1. dereceden Taylor açılımı konur.

$$\frac{\partial \mathcal{E}(\vec{v})}{\partial \vec{v}} \approx -2 \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (A(x, y) - B(x+v_x) - \left[\frac{\partial B}{\partial x} \frac{\partial B}{\partial y} \right] \vec{v}) \cdot \left[\frac{\partial B}{\partial x} \frac{\partial B}{\partial y} \right] \quad (2.67)$$

$A(x, y) - B(x, y)$ miktarı, noktanın zamana bağlı türeviyle açıklanabilir

$$[x \ y]^T : \forall (x, y) \in [p_x - w_x, p_x + w_x] \times [p_y - w_y, p_y + w_y] \quad (2.68)$$

$$\delta I(x, y) \doteq A(x, y) - B(x, y)$$

$\left[\frac{\partial B}{\partial x} \frac{\partial B}{\partial y} \right]$ matrisi sadece görüntü gradyan vektörüdür. Notasyonda küçük bir değişiklik yapılarak;

$$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \doteq \left[\frac{\partial B}{\partial x} \frac{\partial B}{\partial y} \right]^T \quad (2.69)$$

İkinci görüntü $B(x, y)$ 'den bağımsız olarak p noktasının $A(x, y)$ birinci görüntüsünün içindeki $(2w_x + 1) \times (2w_y + 1)$ komşuluğundan direkt olarak I_x ve I_y görüntü türevleri hesaplanabilir.

Eğer merkezi fark operatörü türev için kullanıldıysa , iki türev görüntüsü aşağıdaki ifade gibi olur:

$$\forall (x, y) \in [p_x - w_x, p_x + w_x] \times [p_y - w_y, p_y + w_y]$$

$$I_x(x, y) = \frac{\partial A(x, y)}{\partial x} = \frac{A(x+1, y) - A(x-1, y)}{2} \quad (2.70)$$

$$I_y(x, y) = \frac{\partial A(x, y)}{\partial y} = \frac{A(x, y+1) - A(x, y-1)}{2} \quad (2.71)$$

Pratikte, Sharr operatörü görüntü türevlerini hesaplamak için kullanılır.

Bu yeni notasyonla, denklem 2.67 şöyle yazılabilir:

$$\frac{1}{2} \frac{\partial \mathcal{E}(\vec{v})}{\partial \vec{v}} \approx \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (\nabla I^T \vec{v} - \delta I) \nabla I^T \quad (2.72)$$

$$\frac{1}{2} \left[\frac{\partial \mathcal{E}(\vec{v})}{\partial \vec{v}} \right] \approx \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \left(\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \vec{v} - \begin{bmatrix} \delta I I_x \\ \delta I I_y \end{bmatrix} \right) \quad (2.73)$$

$$G \doteq \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad ve \quad \vec{b} = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} \delta I I_x \\ \delta I I_y \end{bmatrix} \quad (2.74)$$

Denklem 2.73 şöyle yazılabilir:

$$\vec{v}_{opt} = G^{-1} \vec{b} \quad \frac{1}{2} \left[\frac{\partial \mathcal{E}(\vec{v})}{\partial \vec{v}} \right] \approx G \vec{v} - \vec{b} \quad (2.75)$$

Bu nedenle, 2.65. denklemden elde edilen optik akış vektörü şu şekilde yazılabilir:

$$\vec{v}_{opt} = G^{-1} \vec{b} \quad (2.76)$$

Bu ifade sadece G matrisi ters çevrilebilirse geçerlidir. Bu görüntü $A(x,y)$, p noktasının komşusu olan x ve y yönlerinin gradyan bilgisini içermesi anlamına gelir.

Bu denklem standart Lucas Kanade optik akış denklemidir; bu durum sadece (birinci dereceden Taylor açılımlarını geçerli kılmak amacıyla) piksel değiştirmesi küçük olduğunda geçerli olmaktadır. Pratikte, doğru bir çözüm elde etmek için bir çok defa tekrarlama yapılması önemlidir (Newton-Raphson yöntemi).

Algoritmanın tekrarlama versiyonunun detayları için algoritmanın amacı tekrar hatırlanırsa; hata fonksiyonu $\mathcal{E}(\vec{v})$ 'nu minimize eden $\vec{v}$ vektörü bulunmak istenmektedir.

Tekrarlama indeksi, k 'dır ve her ilk tekrarlama başlangıç değeri 1 'dir. Algoritma tekrarlamalı olarak anlatılırsa; genel tekrarlama $k \geq 1$ 'de, $1, 2, \dots, k-1$ tekrarlamalarından elde edilen önceki hesaplamaların, piksel yer değiştirmesi $\vec{v}$ için başlangıç tahmini $\vec{v}^{k-1} = [v_x^{k-1} \ v_y^{k-1}]^T$ sağladığı varsayılır. B_k , başlangıç tahmini $\vec{v}^{k-1}$ 'ye karşılık gelen yeni çevrilmiş görüntüdür:

$$\forall (x, y) \in [p_x - w_x, p_x + w_x] \times [p_y - w_y, p_y + w_y], \quad (2.77)$$

$$B_k(x, y) = B(x + v_x^{k-1}, y + v_y^{k-1})$$

Amaç; hata fonksiyonunu minimize eden artan piksel hareket vektörünü $\vec{\eta}^k = [\eta_x^k \ \eta_y^k]$ hesaplamak olacaktır:

$$\mathcal{E}^k(\vec{\eta}^k) = \mathcal{E}^L(\eta_x^k, \eta_y^k) = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (A(x, y) - B_k(x + \eta_x^k, y + \eta_y^k))^2 \quad (2.78)$$

Bu işlemin çözümü bir Lucas Kanade optik akış hesabı adımı ile hesaplanabilir:

$$\vec{\eta}^k = G^{-1} \vec{b}_k \quad (2.79)$$

burada $2x1$ 'lik vektör $\vec{b}_k$, aşağıdaki denklemde belirlenmektedir (ayrıca görüntü eşleştirme vektörü olarak adlandırılır):

$$\vec{b}_k = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} \delta I_k(x, y) I_x(x, y) \\ \delta I_k(x, y) I_y(x, y) \end{bmatrix} \quad (2.80)$$

burada k 'inci görüntü farkı δI_k aşağıdaki şekilde tanımlanmaktadır.,

$$\forall (x, y) \in [p_x - w_x, p_x + w_x] \times [p_y - w_y, p_y + w_y], \quad \delta I_k(x, y) = A(x, y) - B_k(x, y) \quad (2.81)$$

I_x ve I_y uzaysal türevleri ($\vec{p}$ komşuluğundaki her noktada) sadece bir defa 2.70 ve 2.71 denklemlerinin tekrarlama başlangıcında hesaplanır. Bu nedenle $2x2$ 'lik matris G de tekrarlama çevrimi boyunca sabit kalır (denklem 2.74 ile verilen ifade). Bu, belirgin hesapsal bir avantaj oluşturur. Her adım k 'da tekrar hesaplanması gereken tek değer $\vec{b}_k$ 'dır; bu vektör, $\vec{v}^{k-1}$ vektörü tarafından çevrimden sonra görüntü yamaları arasında artan fark miktarını yakalar. Bir defa artan optik akış $\vec{\eta}^k$ (denklem 2.79 'da) hesaplandıktan sonra, yeni bir piksel yer değiştirmesi $\vec{v}^k$, bir sonraki tekrarlama adımı $k+1$ için hesaplanır:

$$\vec{v}^k = \vec{v}^{k-1} + \vec{\eta}^k \quad (2.82)$$

Tekrarlama sahnesi, hesaplana piksel artanı $\vec{\eta}^k$ bir eşik değerinden daha küçük olana dek veya maksimum tekrarlama sayısına erişene kadar devam eder. Ortalamada, 5 tekrarlama yakınsama için yeterlidir. İlk tekrarlama ($k=1$) başlangıç tahmini sıfıra eşitlenir:

$$\vec{v}^0 = [0 \ 0]^T \quad (2.83)$$

K tekrarlamalarının yakınsamaya erişmesi için önemli olduğu kabulü yapılarak; optik akış vektörü $\vec{v} = d^L$ için final çözüm:

$$\vec{v} = d^L = \vec{v}^K = \sum_{k=1}^K \vec{\eta}^k \quad (2.84)$$

Bu vektör, denklem 2.64 (veya denklem 2.57) 'de tanımlanan hata fonksiyonunu minimize eder. Bu, tekrarlamalı Lucas Kanade optik akış hesabı ifadesini sonlandırır. d^L vektörü, denklem 2.58 'den beslenir ve bu toplam prosedür tüm alt-dizi seviyelerinde $L-1, L-2, \dots, 0$ tekrarlanır.

2.3.5.8 Lucas Kanade Yönteminin Gauss Piramidi Algoritması

Gauss piramidi ile birlikte uygulanan Lucas ve Kanade algoritması genel hatları ile maddeler halinde özetlenmiştir:

Amaç: u , görüntü I 'nın bir noktasıdır. Görüntü J üzerinde ilgili nokta v 'yi bulma.

I ve J 'nin piramit gösterimi yapısı: $\{I^L\}_{L=0, \dots, L_m}$ ve $\{J^L\}_{L=0, \dots, L_m}$

Piramit tahmini başlangıç değeri : $g^{L_m} = [g_x^{L_m} \ g_y^{L_m}]^T = [0 \ 0]^T$

- for $L = L_m$ 'den 0 'a kadar -1 adımlarla;

Görüntü I^L üzerindeki u noktasının konumu: $u^L = [p_x \ p_y]^T = u/2^L$

x 'e bağlı olarak I^L 'nin türevi : $I_x(x, y) = \frac{I^L(x+1, y) - I^L(x-1, y)}{2}$

y 'e bağlı olarak I^L 'nin türevi : $I_y(x, y) = \frac{I^L(x, y+1) - I^L(x, y-1)}{2}$

Uzaysal gradyan matrisi: $G = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} I_x^2(x, y) & I_x(x, y)I_y(x, y) \\ I_x(x, y)I_y(x, y) & I_y^2(x, y) \end{bmatrix}$

L-K yinelemesi başlangıç değeri : $\vec{v} = [0 \ 0]^T$

- for $k = 0$ 'dan K 'ya kadar 1 adım aralıklarla (veya $\|\vec{\eta}^k\| < \text{doğruluk eşik değeri}$);

Görüntü farkı :

$$\delta I_k(x, y) = I^L(x, y) - J^L(x + g_x^L + v_x^{k-1}, y + g_y^L + v_y^{k-1})$$

Görüntü eşleştirme vektörü : $\vec{b}_k = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} \delta I_k(x, y) I_x(x, y) \\ \delta I_k(x, y) I_y(x, y) \end{bmatrix}$

Optik akış (Lucas-Kanade) : $\vec{\eta}^k = G^{-1} \vec{b}_k$

Bir sonraki yineleme için tahmin : $\vec{v}^k = \vec{v}^{k-1} + \vec{\eta}^k$

end (k üzerindeki döngünün sonu);

L seviyesindeki, final optik akış : $d^L = \vec{v}^k$

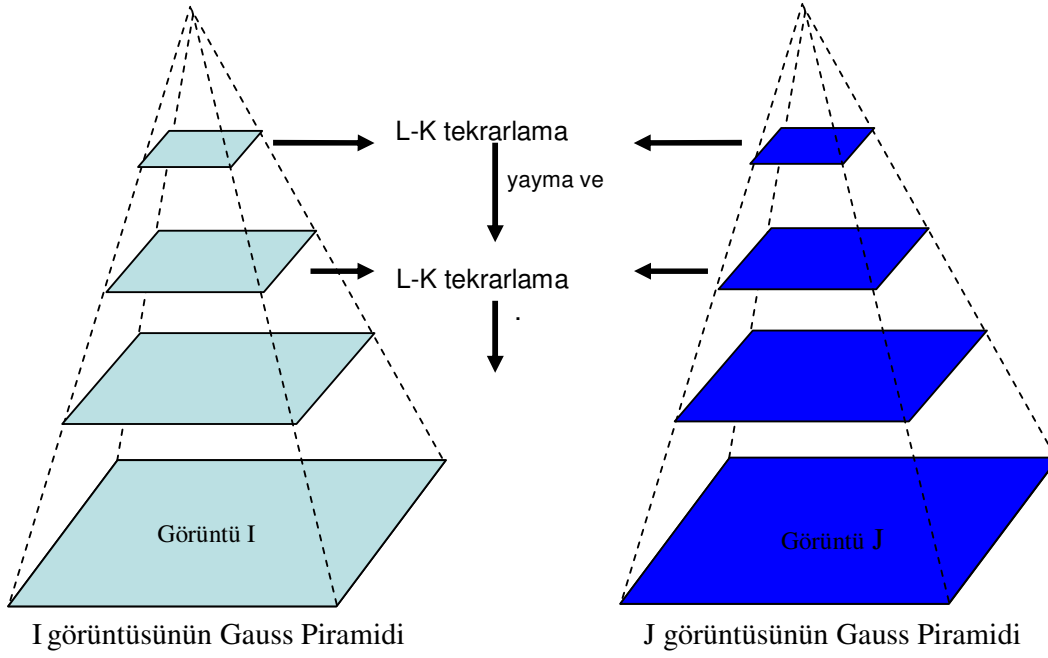
Bir sonraki seviye L-1 için tahmin : $g^{L-1} = [g_x^{L-1} \ g_y^{L-1}]^T = 2(g^L + d^L)$

end (L üzerindeki döngü için son);

Final optik akış vektörü: $d = g^0 + d^0$

Noktanın J üzerindeki konumu: $v = u + d$

Çözüm: J görüntüsü üzerindeki ilgili nokta, v konumundadır.



Şekil 2.21 Görüntünün t ve $t+1$ anındaki iki çerçevesi için Gauss Piramidi gösterimi.

2.3.5.9 Test Sonuçları

Bu bölümde Gauss piramidi ile birlikte uygulanan Lucas ve Kanade yönteminin MATLAB uygulaması sonuçları gösterilmiştir. Burada öncelikle yapılması gereken işlem görüntünün çözünürlüğünü düşürerek farklı boyutlarda görüntü dizileri oluşturmaktır. Elde edilen en küçük boyuttaki görüntü dizisine Lucas ve Kanade algoritması uygulanır ve hız tahmini çıkışları bir sonraki piramit seviyesindeki görüntü dizisine hız tahmini düzenlemek için verilir. Burada uygulanan maske boyutu haricinde, hem belirlenen piramit seviye sayısı hem de algoritmanın tekrarlama sayısı hız tahmininde önemli rol oynamaktadır.



Şekil 2.22 Orijinal görüntü dizileri (Hamburg Taksisi).



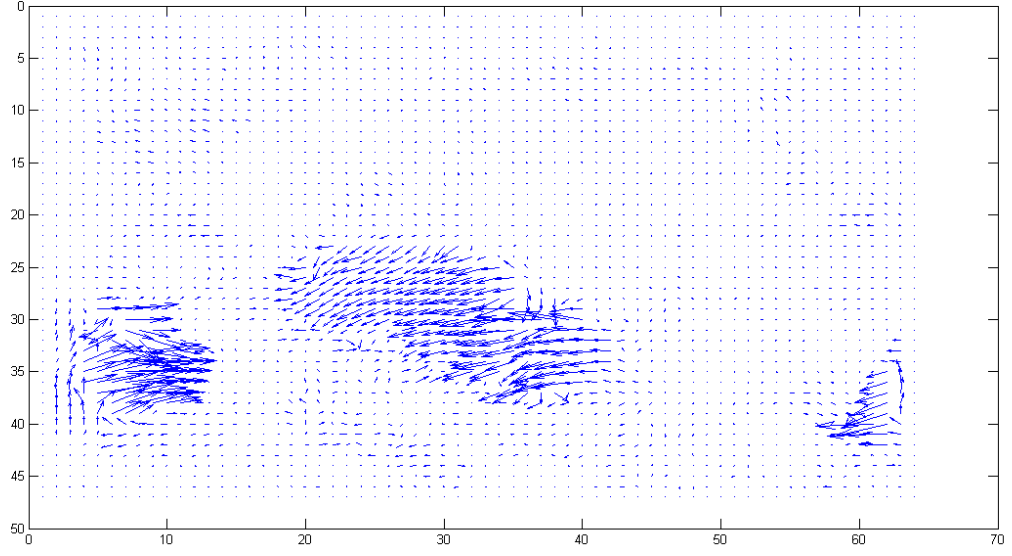
Şekil 2.23 (a) En üst piramit seviyesinde yer alan temel görüntü dizileri.



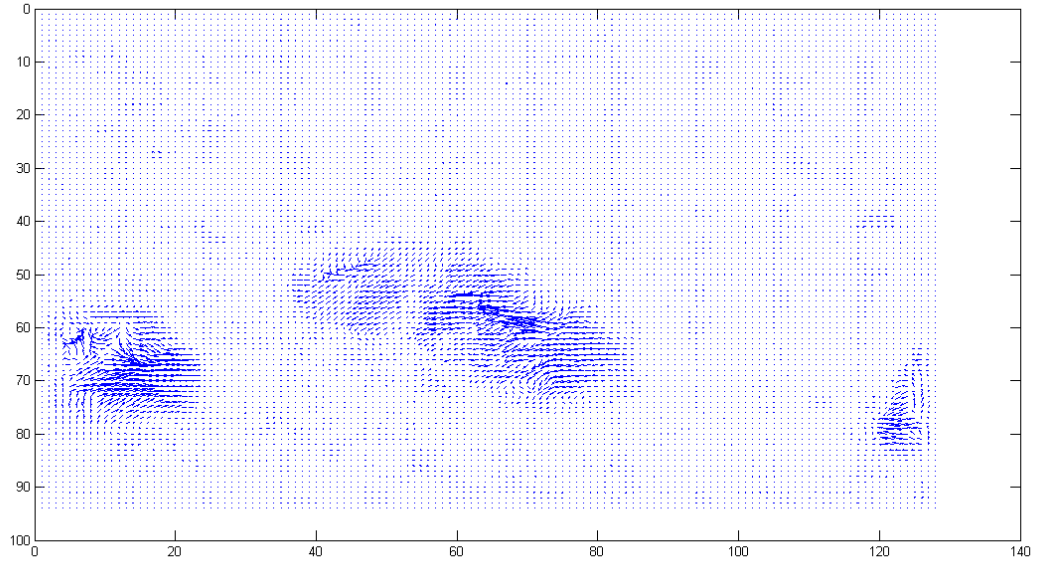
Şekil 2.23 (b) İkinci piramit seviyesinde yer alan görüntü dizileri.



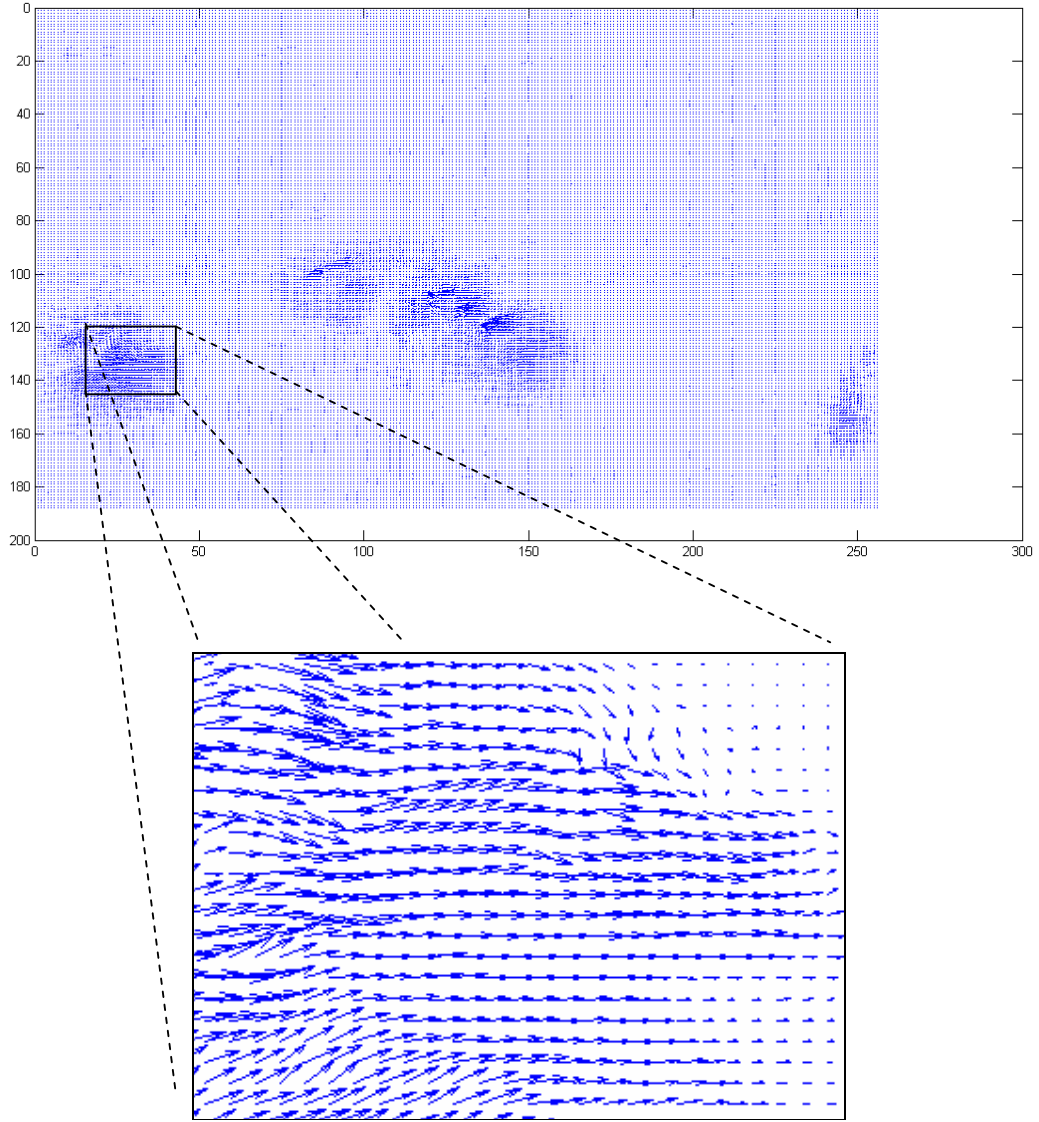
Şekil 2.23 (c) En alt piramit seviyesinde yer alan görüntü dizileri.



Şekil 2.24 (a) En üst piramit seviyesinden elde edilen hız vektörleri.



Şekil 2.24 (b) İkinci piramit seviyesinden elde edilen hız vektörleri.



Şekil 2.24 (c) En alt piramit seviyesinden elde edilen hız vektörleri.

2.3.6 Lucas Kanade Uygulamalarında Karşılaşılan Hatalar

Lucas ve Kanade algoritmasının uygulamalarında karşılaşılan hataların nedenleri arasında $A^T A$ 'nın kolayca dönüştürülebilir olduğu ve görüntüde çok fazla gürültü olmadığı varsayımları yer almaktadır. Şiddet değişmezliği sağlanmıyorsa, hareket değişimi örnekleme hızından daha küçük değilse veya bir nokta komşularına benzer şekilde ilerlemiyorsa bu varsayımlar karşılanamaz ve Lucas ve Kanade algoritması doğru sonuçlar üretemez duruma gelir. Bu nedenle Lucas ve Kanade algoritmasında kullanılan pencere büyüklüğü bu varsayımlar dikkate alınarak belirlenmelidir.

3. SONUÇLAR ve ÖNERİLER

Hareket kısıtlama denklemi tek başına hareket tahmini yapılmasında yeterli olmadığı için ek bileşenler ortaya konarak elde edilen optik akış algoritmaları incelenmiştir. Optik akış yöntemlerinden genel ve yerel fark yaklaşımları üzerinde durulmuştur. Gerçek görüntü dizileri kullanılarak bahsedilen fark yaklaşımlarının MATLAB uygulamaları gerçekleştirilmiştir.

Gerçeklenen Horn ve Schunck MATLAB uygulaması sonucunda, tekrarlama sayısı ve eşik değeri seçiminin nesnelerin doğru hız tahmininde önemli rol oynadığı görülmüştür.

Lucas ve Kanade yaklaşımı, genel yumuşatma bileşenlerini kullanan Horn ve Schunck yöntemine karşılık hesaplama açısından daha verimlidir.

Lucas ve Kanade yönteminin Gauss piramidi uygulaması; daha büyük değişimlerde algoritmanın daha iyi sonuçlar verdiği gösterilmiştir.

Fark tekniklerinde karşılaşılan ana sorunlar arasında süreksizlik yer almaktadır. Fark teknikleri süreksiz nesneleri (örneğin, hareketi sırasında başka bir nesne tarafından örtülen nesneler) istenildiği gibi elde edemez ve bu nedenle hız tahmini algoritmaları tek başına yetersiz kalır. Bu tür sorunlar için nesne tanıma gibi ek algoritmalar kullanılması gerekmektedir.

Bu tez çalışmasında incelenen Hamburg Taksi görüntü dizisi uygulamalarında, sahnede birden fazla farklı yönlerde hareket eden nesne bulunmaktadır. Bu nesneler görüş alanı içerisinde aynı derinlikte hareket etmemeleri nedeniyle hız tahmini algoritmasına ek olarak derinlik tahmini de göz önüne alınırsa daha iyi sonuçlar elde edilecektir. Ayrıca hız tahmini algoritmalarında piksellerin parlaklık genliği temel alındığından gürültü etkisi yumuşatma yapılarak giderilmelidir fakat bu çözüm de küçük hareketlerin algılanmasını engelleyecektir.

KAYNAKLAR

- [1] Berthold Klaus Paul Horn, "Robot Vision" Chapter 12, MIT Press, 1986.
- [2] B.D. Lucas ve T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", DARPA Image Understanding Workshop, 1981, pp121-130 (see also IJCAI'81, pp674-679).
- [3] B.K.P. Horn ve B.G. Schunck, "Determining Optical Flow", Artificial Intelligence, 17, 1981, pp185-204.
- [4] J.L. Barron, D.J. Fleet ve S.S. Beauchemin, "Performance of Optical Flow Techniques", Int. Journal of Computer Vision, 12, 1994, pp. 43-77.
- [5] J.L.Barron and N.A.Thacker, "Tutorial: Computing 2D and 3D Optical Flow", Tina Memo No. 2004-012.
- [6] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm", Intel Corporation, Microprocessor Research Labs. OpenCV Documents, 1999.
- [7] B. D. Lucas, "Generalized Image Matching by the Method of Differences", doctoral dissertation, 1984.
- [8] Edward H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, ve J. M. Ogden, "Pyramid Method in Image Processing", published in RCA Engineer, vol. 29(6), pp. 33-41, 1984.
- [9] Marcus Nyström, "Novelty and Change Detection" KTH, Haziran 2003
- [10] T. Lindeberg, "Scale-Space Theory in Computer Vision", Kluwer, 1994.

ÖZGEÇMİŞ

Doğum Tarihi	25.09.1981	
Doğum yeri	Gölcük	
Lise	1995-1999	Pendik Lisesi (İng.)
Lisans	2000-2004	Kocaeli Üniversitesi Elektrik-Elektronik Fakültesi Elektronik ve Haberleşme Mühendisliği
Yüksek Lisans	2004-2007	Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Müh. Anabilim Dalı Haberleşme Programı

Çalıştığı Kurum(lar)

2005-2006	Berk Bilişim ve İletişim Sist. LTD. ŞTİ. Ürün Yöneticisi
2006- ...	NORTEL NETAŞ Telekomünikasyon A.Ş. Yazılım Destek Mühendisi