Chapter 1

Introduction

1.1 Introduction

Three problems in vision:

- Perceptual Organization: Grouping and Figure/Ground (more broadly, contour classification)
- Inference of surface properties: depth, orientation, lightness, color, movement (more broadly, the inverse optics program)
- Categories and Affordances (recognition of scenes, objects, materials, activities, functionalities etc)

What information do we need to solve this problem?
The obvious one is optics:

- Process of image formation
- Properties of light transport
- Geometry and simple physics

However using optics alone is insufficient. Equally important are constraints derived from the actual world in which we live. Gibson articulated these in his notion of "Ecological Optics". The environment consists of the surrounds of animals. The environment is not the same as the physical world, if one means by that the world described by physics. Rather it is the relevant structure, at spatial and temporal scales corresponding to the animal. The terrestrial environment contains

- Surfaces that have a certain layout.
- The Ground is the surface of the earth.
- Objects are persisting substances with closed or nearly closed surfaces. Could be detached or attached.
- Enclosures are layouts of surfaces that surround the medium to some degree.
- Places that are locations in the environment. The habitat of an animal is made up of places.

These constraints are probabilistic nature and can be captured with statistical models.

Let us consider the relationship of computer vision to the three paradigms for studying biological vision:

1. Perception: study the "laws of seeing" i.e predict what a human would perceive in an image.
2. Neuroscience: understand the mechanisms in the retina and the brain
3. Function: how laws of optics, and the statistics of the world we live in, make certain interpretations of an image more likely to be valid

The match between human and computer vision is strongest at the level of function, but since typically the results of computer vision are meant to be conveyed to humans makes it useful to be consistent with human perception. Neuroscience is a source of ideas but being bio-mimetic is not a requirement.

Chapter 2

Geometry of Image Formation

There are two fundamental questions related to image formation:

- Where is a point in the world imaged?
- How bright is the resulting image point?

We start with the first question, for which it is adequate to use the **pinhole camera** model. Historically, this originated with the camera obscura. That the image was inverted confused people for quite some time and delayed the application of this model to image formation in the retina. Finally, Kepler in 1604, Descartes in 1620s experimentally showed that the image really is inverted, there was no system of mirrors or lenses in the eye that made the image the right side up.

An understanding of the basic mathematics of perspective precedes Kepler and Descartes. It goes back to Euclid, Alhazen, and of course the painters of the Italian Renaissance. While credit for the first artistic creations is given to Brunelleschi and Masaccio, the first formal statement of the principles is usually attributed to Alberti (1435).

2.1 Perspective Projection

A pinhole camera consists of a pinhole opening, O, at the front of a box, and an image plane at the back of the box , see Figure 2.1. We will use a three-dimensional coordinate system with the origin at O and will consider a point P in the scene, with coordinates (X, Y, Z). P gets projected to the point P' in the image plane with coordinates (x, y, z). If f is the distance from the pinhole to the image plane, then by similar triangles, we can derive the following equations:

$$\frac{-x}{f} = \frac{X}{Z}, \; \frac{-y}{f} = \frac{Y}{Z} \quad \Rightarrow \quad x = \frac{-fX}{Z}, \; y = \frac{-fY}{Z} \; .$$

These equations define an image formation process known as perspective projection. Note that the Z in the denominator means that the farther away an object is, the smaller its image will be. Also, note that the minus signs mean that the image is *inverted*, both left–right and up–down, compared with the scene.

Equivalently, we can model the perspective projection process with the projection plane being at a distance f in *front* of the pinhole. This device of imagining a projection surface in front was first recommended to painters in the Italian Renaissance by Alberti in 1435 as a technique for constructing geometrically accurate depictions of a three-dimensional scene. For our purposes, the main advantage of this model is that it avoids lateral inversion and thereby eliminates the negative signs in the perspective projection equations. Note that it isn't essential that the projection surface be a plane, it could equally well be a sphere centered at the pinhole. The key aspect here is the 1-1 mapping from rays through the pinhole to points on a projection surface.

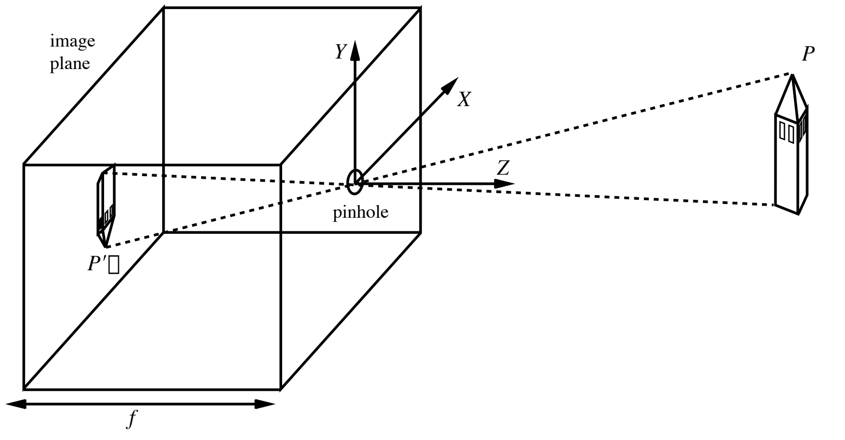


Figure 2.1: Pinhole Camera.

Canonically, in spherical perspective, the projection surface is a sphere of unit radius ("viewing sphere") centered at the center of projection. A point (ρ, θ, ϕ) gets mapped to $(1, \theta, \phi)$. The ray from the point in the scene through the center of projection is perpendicular to the imaging surface. Spherical perspective avoids an artifact of plane perspective known as the "position effect".

Let's summarize the preceding discussion in vector notation. A world point $\mathbf{X}$ projects to a point on a plane, $\mathbf{p}$, under planar projection, and equivalently a point on a sphere, $\mathbf{q}$ under spherical projection:

$$\begin{aligned}\mathbf{p} &= f\frac{\mathbf{X}}{Z}\\ \mathbf{q} &= \frac{\mathbf{X}}{\|\mathbf{X}\|}\end{aligned}$$

where $\mathbf{p} = (x, y, f)$, and $\mathbf{q}$ is a unit vector. Note, precisely the same information is represented in each case - namely the ray *direction* which is the most that can be recovered from an image point. It is straightforward to transform between $\mathbf{p}$ and $\mathbf{q}$: $\mathbf{q} = \mathbf{p}/||\mathbf{p}||$, $\mathbf{p} = f\mathbf{q}/q_z$. The connection between the vector notation and a planar image is that if $\mathbf{p} = (x, y, f)$, then $\mathbf{x} = (x, y)$.

2.2 Projection of lines

A line of points in 3D can be represented as $\mathbf{X} = \mathbf{A} + \lambda\mathbf{D}$, where $\mathbf{A}$ is a fixed point, $\mathbf{D}$ a unit vector parallel to the line, and λ a measure of distance along the line. As λ increases points are increasingly further away and in the limit:

$$\lim_{\lambda\to\infty} \mathbf{p} = f\frac{\mathbf{A} + \lambda\mathbf{D}}{A_Z + \lambda D_Z} = f\frac{\mathbf{D}}{D_Z}$$

i.e. the image of the line terminates in a *vanishing point* with coordinates $(fD_X/D_Z, fD_Y/D_Z)$, unless the line is parallel to the image plane ($D_Z = 0$). Note, the vanishing point is unaffected (invariant to) line position, $\mathbf{A}$, it only depends on line orientation, $\mathbf{D}$. Consequently, the family of lines parallel to $\mathbf{D}$ have the same vanishing point.

Under spherical perspective, a line in the scene projects to half of a great circle. This circle is defined by the intersection of the viewing sphere with the plane containing the line and center of projection. There are two vanishing points here corresponding to the endpoints of the half great circle. You should convince yourself that these are the same for a family of parallel straight lines.

2.3 Projection of planes

A plane of points in 3D can be represented as $\mathbf{X}.\mathbf{N} = d$ where $\mathbf{N}$ is the unit plane normal, and d the perpendicular distance of the plane from the origin. A point $\mathbf{X}$ on the plane is imaged at $\mathbf{p} = f\mathbf{X}/Z$. Taking the scalar product of both sides with $\mathbf{N}$ gives $\mathbf{p}.\mathbf{N} = f\mathbf{X}.\mathbf{N}/Z = fd/Z$. In the limit of points very distant:

$$\lim_{Z\to\infty} \mathbf{p}.\mathbf{N} = 0$$

which is the equation of a plane through the origin parallel to the world plane (i.e. which has the same normal $\mathbf{N}$). The plane $\mathbf{p}.\mathbf{N} = 0$ intersects the image plane in a *vanishing line*

$$xN_x + yN_y + fN_z = 0$$

Note, the vanishing line is unaffected (invariant to) plane position, d, it only depends on plane orientation, $\mathbf{N}$. All planes with the same orientation have the same vanishing line, also called the horizon.

Consider a line on the plane. It can be shown (exercise) that the vanishing points of all lines on the plane lie on the vanishing line of the plane. Thus, two vanishing points determine the vanishing line of the plane.

Under spherical perspective, the horizon of a plane is a great circle, found by translating the plane parallel to itself until it passes through the center of projection, and then intersecting it with the viewing sphere.

2.4 Terrestrial Perspective

Consider an observer standing on a ground plane looking straight ahead of her. Since the ground plane has surface normal $\mathbf{N} = (0, 1, 0)$, the equation of the horizon is $y = 0$. In this canonical case, the horizon lies in the middle of the field of view, with the ground plane in the lower half and the sky in the upper half.

Let us work out how objects of different heights and at different locations on the ground plane project. We will suppose that the eye, or camera, is a height h_c above the ground plane. Consider an object of height δY resting on the ground plane, whose bottom is at $(X, -h_c, Z)$ and top is at $(X, \delta Y - h_c, Z)$. The bottom projects to $(fX/Z, -fh_c/Z)$ and the top to $(fX/Z, f(\delta Y - h_c)/Z)$.

We note the following:

1. The bottoms of nearer objects (small Z) project to points lower in the image plane, farther objects have bottoms closer to the horizon.

2. If the object has the same height as the camera ($\delta Y = h_c$), the projection of its top lies on the horizon.

3. The ratio of the height of the object to the height of the camera, $\delta Y/h_c$ is the ratio of the apparent vertical height of the object in the image to the vertical distance of its bottom from the horizon (Verify).

2.5 Orthographic Projection

If the object is relatively shallow compared with its distance from the camera, we can approximate perspective projection by scaled orthographic projection. The idea is as follows: If the depth Z of points on the object varies within some range $Z_0 \pm \Delta Z$, with $\Delta Z \ll Z_0$, then the perspective scaling factor f/Z can be approximated by a constant $s = f/Z_0$. The equations for projection from the scene coordinates (X, Y, Z) to the image plane become $x = sX$ and $y = sY$. Note that scaled orthographic projection is an approximation that is valid only for those parts of the scene with not much internal depth variation; it should not be used to study properties "in the large." For instance, under orthographic projection, parallel lines stay parallel instead of converging to a vanishing point!

2.6 Summary

- Plane perspective

$$(X, Y, Z) \mapsto (\frac{fX}{Z}, \frac{fY}{Z}, f) \tag{2.1}$$

- Spherical perspective

$$(X, Y, Z) \mapsto (\frac{X, Y, Z}{\sqrt{X^2 + Y^2 + Z^2}}) \tag{2.2}$$

- Lines $\mapsto$ vanishing points

$$A + \lambda D \mapsto (\frac{fD_x}{D_z}, \frac{fD_y}{D_z}) \tag{2.3}$$

- Planes $\mapsto$ vanishing lines (horizons)

$$X \cdot N = d \mapsto xN_x + yN_y + fN_z = 0 \tag{2.4}$$

2.7 Exercises

1. Show that the vanishing points of lines on a plane lie on the vanishing line of the plane.

2. Show that, under typical conditions, the silhouette of a sphere of radius r with center (X,0, Z) under planar perspective projection is an ellipse of eccentricity $X/\sqrt{(}X^2 + Z^2 - r^2)$. Are there circumstances under which the projection could be a parabola or hyperbola? What is the silhouette for spherical perspective?

3. An observer is standing on a ground plane looking straight ahead. We want to calculate the accuracy with which she will be able to estimate the depth Z of points on the ground plane, assuming that she can visually discriminate angles to within $1'$. Derive a formula relating depth error δZ to Z. For simplicity, just consider points straight ahead of the observer($x = 0$). Given a Z value (say 10 m), your formula should be able to predict the δZ.

Chapter 3

Pose, Shape and Geometric Transformations

Points on an object can be characterized by their 3D coordinates with respect to the camera coordinate system. But what happens, when we move the object? In a certain sense when a chair is moved in 3D space, it remains the "same" even though the coordinates of points on it with respect to the camera (or any fixed) coordinate system do change. This distinction is captured by the use of the terms **pose** and **shape**.

- *Pose:* The position and orientation of the object with respect to the camera. This is specified by 6 numbers (3 for its translation, 3 for rotation). For example, we might consider the coordinates of the centroid of the object relative to the center of projection, and the rotation of a coordinate frame on the object with respect to that of the camera.

- *Shape:* The coordinates of the points of an object relative to a coordinate frame on the object. These remain invariant when the object undergoes rotations and translations.

To make these notions more precise, we need to develop the basic theory of **Euclidean Transformations**. The set of transformations defines a notion of "congruence" or having the same shape. In high school geometry we learned that two planar triangles are congruent if one of them can be rotated and translation so as to lie exactly on top of another. Rotation and translation are examples of Euclidean transformations, also known as **isometries** or **rigid body motions**, defined as transformation that preserve distances between any pair of points. When I move a chair, this holds true between any pair of points on the chair, but obviously not for points on a balloon that is being inflated.

In this chapter we will review the basic concepts relevant to Euclidean transformations. Then we will study a more general class of transformations, called **affine transformations**, which include Euclidean transformations as a subset. The set of **projective transformations** is even more general, and is a superset of affine transformations. All three classes of transformations find utility in a study of vision.

3.1 Euclidean Transformations

$\mathbf{A}$	Matrix
$\mathbf{a}$	Vector
$\mathbf{I}$	The identity matrix
$\psi : \mathbb{R}^n \mapsto \mathbb{R}^n$	Transformation
$\mathbf{x} \cdot \mathbf{y}$	Dot product (scalar product)
$\mathbf{x} \wedge \mathbf{y}$	Cross product (vector product)
$\|\|\mathbf{x}\|\| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$	Norm

Definition 1 *Euclidean transformations (also known as isometries) are transformations that preserve distances between pairs of points.*

$$||\psi(\mathbf{a}) - \psi(\mathbf{b})|| = ||\mathbf{a} - \mathbf{b}|| \tag{3.1}$$

Translations, $\psi(\mathbf{a}) = \mathbf{a} + \mathbf{t}$, are isometries, since

$$||\psi(\mathbf{a}) - \psi(\mathbf{b})|| = ||\mathbf{t} + \mathbf{a} - (\mathbf{t} + \mathbf{b})|| = ||\mathbf{a} - \mathbf{b}|| \tag{3.2}$$

We now define orthogonal transformations; these constitute another major class of isometries.

Definition 2 *A linear transformation: $\psi(\mathbf{a}) = \mathbf{A}\mathbf{a}$, for some matrix $\mathbf{A}$.*

Definition 3 *Orthogonal transformations are linear transformations which preserve inner products.*

$$\mathbf{a} \cdot \mathbf{b} = \psi(\mathbf{a}) \cdot \psi(\mathbf{b}) \tag{3.3}$$

Property 1 *Orthogonal transformations preserve norms.*

$$\mathbf{a} \cdot \mathbf{a} = \psi(\mathbf{a}) \cdot \psi(\mathbf{a}) \implies ||\mathbf{a}|| = ||\psi(\mathbf{a})|| \tag{3.4}$$

Property 2 *Orthogonal transformations are isometries.*

$$(\psi(\mathbf{a}) - \psi(\mathbf{b})) \cdot (\psi(\mathbf{a}) - \psi(\mathbf{b})) \overset{?}{=} (\mathbf{a} - \mathbf{b}) \cdot (\mathbf{a} - \mathbf{b}) \tag{3.5}$$

$$||\psi(\mathbf{a})||^2 + ||\psi(\mathbf{b})||^2 - 2(\psi(\mathbf{a}) \cdot \psi(\mathbf{b})) \overset{?}{=} ||\mathbf{a}||^2 + ||\mathbf{b}||^2 - 2(\mathbf{a} \cdot \mathbf{b}) \tag{3.6}$$

By property 1,

$$||\psi(\mathbf{a})||^2 = ||\mathbf{a}||^2 \tag{3.7}$$

$$||\psi(\mathbf{b})||^2 = ||\mathbf{b}||^2. \tag{3.8}$$

By definition 3,

$$\psi(\mathbf{a}) \cdot \psi(\mathbf{b}) = \mathbf{a} \cdot \mathbf{b}. \tag{3.9}$$

Thus, equality holds.

Note that translations do not preserve norms (the distance with respect to the origin changes) and are not even linear transformations, except for the trivial case of translation by $\mathbf{0}$.

3.1.1 Properties of orthogonal matrices

Let ψ be an orthogonal transformation whose action we can represent by matrix multiplication, $\psi(\mathbf{a}) = \mathbf{A}\mathbf{a}$. Then, because it preserves inner products:

$$\psi(\mathbf{a}) \cdot \psi(\mathbf{b}) = \mathbf{a}^T \mathbf{b}\,. \tag{3.10}$$

By substitution,

$$\begin{aligned} \psi(\mathbf{a}) \cdot \psi(\mathbf{b}) &= (\mathbf{A}\mathbf{a})^T(\mathbf{A}\mathbf{b}) && (3.11)\\ &= \mathbf{a}^T\mathbf{A}^T\mathbf{A}\mathbf{b}\,. && (3.12)\end{aligned}$$

Thus,

$$\mathbf{a}^T\mathbf{b} = \mathbf{a}^T\mathbf{A}^T\mathbf{A}\mathbf{b} \implies \mathbf{A}^T\mathbf{A} = \mathbf{I} \implies \mathbf{A}^T = \mathbf{A}^{-1}\,. \tag{3.13}$$

Note that $\det(\mathbf{A})^2 = 1$ which implies that $\det(\mathbf{A}) = +1$ or -1. Each column of $\mathbf{A}$ has norm 1, and is orthogonal to the other column.

In 2D, these constraints put together force $\mathbf{A}$ to be one of two types of matrices.

$$\underbrace{\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}}_{\text{rotation, det=+1}} \text{or} \underbrace{\begin{bmatrix}\cos\theta & \sin\theta\\ \sin\theta & -\cos\theta\end{bmatrix}}_{\text{reflection, det=}-1}$$

Under a rotation by angle θ,

$$\begin{bmatrix}1\\0\end{bmatrix} \mapsto \begin{bmatrix}\cos\theta\\ \sin\theta\end{bmatrix} \text{and} \begin{bmatrix}0\\1\end{bmatrix} \mapsto \begin{bmatrix}-\sin\theta\\ \cos\theta\end{bmatrix}$$

The reflection matrix above corresponds to reflection around the line with angle $\frac{\theta}{2}$ (verify). Note that two rotations one after the other give another rotation, while two reflections give us a rotation.

Let us now construct some examples in 3D. Just as in 2D, rotations are characterized by orthogonal matrices with det $= +1$. For orthogonal matrices, each column vector has length 1, and the dot product of any two different columns is 0. This gives rise to six constraints (3 pairwise dot product constraints, and 3 length constraints), so for a 3 dimensional rotation matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix} \tag{3.14}$$

with 9 total parameters, there are really only three free parameters. There are several methods by which these parameters can be specified, as we will study later. Here are a few example rotation matrices.

- Rotation about z-axis by θ:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \tag{3.15}$$

- Rotation about x-axis by θ:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{3.16}$$

3.1.2 Group structure of isometries

Theorem 3.1 *Any isometry can be expressed as the combination of an orthogonal transformation followed by a translation as follows:*

$$\psi(\mathbf{a}) = \mathbf{A}\mathbf{a} + \mathbf{t} \tag{3.17}$$

where $\mathbf{A}$ *represents the orthogonal matrix and* $\mathbf{t}$ *is the translation vector.*

The set of rigid body motions constitutes a *group*[1]. In our notation, $\psi_1 \circ \psi_2$, ψ_1 composed with ψ_2, denotes that we apply ψ_2 first and then ψ_1.

We will show first that isometries are closed under composition. Consider two rigid body motions, ψ_1 and ψ_2:

$$\psi_1(\mathbf{a}) = \mathbf{A}_1\mathbf{a} + \mathbf{t}_1 \qquad \psi_2(\mathbf{a}) = \mathbf{A}_2\mathbf{a} + \mathbf{t}_2. \tag{3.18}$$

Then we have

$$\begin{aligned} \psi_1 \circ \psi_2(\mathbf{a}) &= \mathbf{A}_1(\mathbf{A}_2\mathbf{a} + \mathbf{t}_2) + \mathbf{t}_1 & (3.19)\\ &= \mathbf{A}_1\mathbf{A}_2\mathbf{a} + \mathbf{A}_1\mathbf{t}_2 + \mathbf{t}_1 & (3.20)\\ &= (\mathbf{A}_1\mathbf{A}_2)\mathbf{a} + (\mathbf{A}_1\mathbf{t}_2 + \mathbf{t}_1) & (3.21)\\ &= \mathbf{A}_3\mathbf{a} + \mathbf{t}_3 & (3.22) \end{aligned}$$

where $\mathbf{A}_3 = \mathbf{A}_1\mathbf{A}_2$ and $\mathbf{t}_3 = \mathbf{A}_1\mathbf{t}_2 + \mathbf{t}_3$. Thus, $\psi_1 \circ \psi_2 = \psi_3$ is also a rigid body motion, under the assumption that the product of two orthogonal matrices is orthogonal (Verify!)

Note that translations and rotations are closed under composition, but reflections are not.

We can verify the remaining axioms for showing that isometries constitute a group

- Identity: $\mathbf{A} = \mathbf{I}$, $\mathbf{d} = 0$.
- Inverse: We need $\mathbf{A}_1\mathbf{A}_2 = \mathbf{I}$ and $\mathbf{t}_3 = \mathbf{A}_1\mathbf{t}_2 + \mathbf{t}_1 = 0$. This means that for ψ_1 to be the inverse of ψ_2, $\mathbf{A}_1 = \mathbf{A}_2^T$ and $\mathbf{d}_2 = -\mathbf{A}_1^{-1}\mathbf{t}_1$
- Associativity: left as an exercise for the reader.

[1]A group $(G, \circ)$ is a set G with a binary operation $\circ$ that satisfies the following four axioms: Closure: For all a, b in G, the result of $a \circ b$ is also in G. Associativity: For all a, b and c in G, $(a \circ b) \circ c = a \circ (b \circ c)$. Identity element: There exists an element e in G such that for all a in G, $e \circ a = a \circ e = a$. Inverse element: For each a in G, there exists an element b in G such that $a \circ b = b \circ a = e$, where e is an identity element.

3.2 Parametrizing Rotations in 3D

Recall that rotation matrices have the property that each column vector has length 1 and the dot product of any 2 different columns is 0. These 6 constraints leave only 3 degrees of freedom. Here are some alternative notations used to represent orthogonal matrices in 3-D:

- Euler angles which specify rotations about 3 axes
- Axis plus amount of rotation
- Quaternions which generalize complex numbers from 2-D to 3-D. (Note, a complex number can represent a rotation in 2-D)

We will use the axis and rotation as the preferred representation of an orthogonal matrix: $\mathbf{s}$, θ, where $\mathbf{s}$ is the unit vector of the axis of rotation and θ is the amount of rotation.

Definition 4 *A matrix* $\mathbf{S}$ *is skew-symmetric if* $\mathbf{S} = -\mathbf{S}^T$*.*

Skew symmetric matrices can be used to represent "cross" products or vector products. Recall:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \wedge \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix}$$

We define $\widehat{\mathbf{a}}$ as:

$$\widehat{\mathbf{a}} \overset{def}{=} \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

Thus, multiplying $\widehat{\mathbf{a}}$ by any vector gives:

$$\begin{aligned} \widehat{\mathbf{a}} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} &= \begin{bmatrix} -a_3b_2 + a_2b_3 \\ a_3b_1 - a_1b_3 \\ -a_2b_1 + a_1b_2 \end{bmatrix} \\ &= \mathbf{a} \wedge \mathbf{b} \end{aligned}$$

Consider now, the equation of motion of a point q on a rotating body:

$$\dot{\mathbf{q}}(t) = \omega \wedge \mathbf{q}(t)$$

where the direction of ω specifies the axis of rotation and $\|\omega\|$ specifies the angular speed. Rewriting with $\widehat{\omega}$

$$\dot{\mathbf{q}}(t) = \widehat{\omega}\mathbf{q}(t)$$

The solution of this differential equation involves the exponential of a matrix. (In matlab, this is the operator `expm`.)

$$\mathbf{q}(t) = e^{\widehat{\omega}t}\mathbf{q}(0)$$

Where,

$$e^{\widehat{\omega}t} = \mathbf{I} + \widehat{\omega}t + \frac{(\widehat{\omega}t)^2}{2!} + \frac{(\widehat{\omega}t)^3}{3!} + ...$$

Collecting the odd and even terms in the above equation, we get to **Roderigues Formula** for a rotation matrix $\mathbf{R}$.

$$\begin{aligned}\mathbf{R} &= e^{\phi\widehat{s}} \\ &= \mathbf{I} + \sin\phi\ \widehat{\mathbf{s}} + (1-\cos\phi)\widehat{\mathbf{s}}^2\end{aligned}$$

Here $\mathbf{s}$ is a unit vector along ω and $\phi = \|\omega\|t$ is the total amount of rotation. Given an axis of rotation, $\mathbf{s}$, and amount of rotation ϕ we can construct $\widehat{\mathbf{s}}$ and plug it in.

3.3 Affine transformations

Thus far we have focused on Euclidean transformations, $\psi(\mathbf{a}) = \mathbf{A}\mathbf{a} + \mathbf{t}$, where $\mathbf{A}$ is an orthogonal matrix. If we allow $\mathbf{A}$ to be any non-singular matrix (i.e., $\det\mathbf{A} \neq 0$), then we get the set of affine transformations. Note that the Euclidean transformations are a subset of the affine transformations.

3.3.1 Degrees of freedom

Let us count the degrees of freedom in the parameters that specify a transformation. For $\psi : \mathbb{R}^2 \mapsto \mathbb{R}^2$, Euclidean transformations have 3 free parameters (1 rotation, 2 translation), whereas Affine transformations have 6 (4 in $\mathbf{A}$ and 2 in $\mathbf{t}$). For $\psi : \mathbb{R}^3 \mapsto \mathbb{R}^3$, Euclidean transformations have 6 free parameters (3 rotation, 3 translation), whereas Affine transformations have 12 (9 in $\mathbf{A}$ and 3 in $\mathbf{t}$).

3.4 Exercises

1. Show that in $\mathcal{R}^2$ reflection about the $\theta = \alpha$ line followed by reflection about the $\theta = \beta$ is equivalent to a rotation of $2(\beta - \alpha)$.

2. Verify Roderigues formula by considering the powers of the skew-symmetric matrix associated with the cross product with a vector.

3. Write a Matlab function for computing the orthogonal matrix $\mathbf{R}$ corresponding to rotation ϕ about the axis vector $\mathbf{s}$. Find the eigenvalues and eigenvectors of the orthogonal matrices and study any relationship to the axis vector. Verify the formula $\cos\phi = \frac{1}{2}\{\text{trace}(\mathbf{R}) - 1\}$. Show some points before and after the rotation has been applied.

4. Write a Matlab function for the converse of that in the previous problem i.e. given an orthogonal matrix $\mathbf{R}$, compute the axis of rotation $\mathbf{s}$ and ϕ). Hint: Show that $\mathbf{R} - \mathbf{R}^T = (2\sin\phi)\widehat{\mathbf{s}}$

Chapter 4

Dynamic Perspective

4.1 Optical Flow

Motion in the 3D world, either of objects or of the camera, projects to motion in the image. We call this **optical flow**. At every point (x, y) in the image we get a 2D vector, corresponding to the motion of the feature located at that point. Thus optical flow is a 2D vector field. As first pointed out by Gibson, the optical flow field of a moving observer contains information to infer the 3D structure of the scene, as well as the movement of the observer, so-called **egomotion**. An example flow field is shown in Figure 4.1.

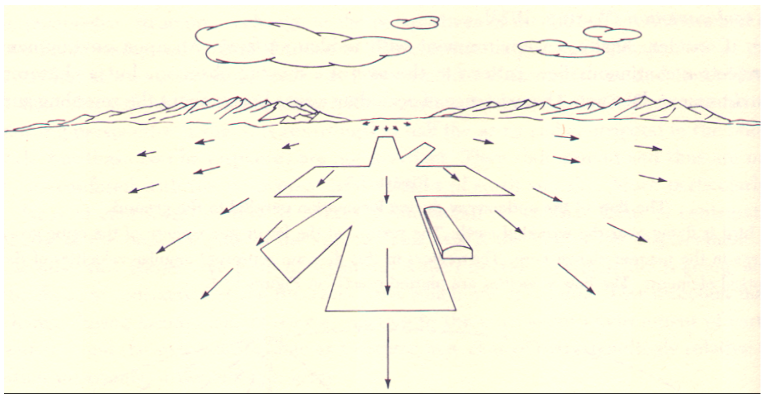


Figure 4.1: The optical flow field of a pilot just before takeoff

4.2 From 3-D Motion to 2-D Optical Flow

- $\mathbf{X} = (X, Y, Z)$: 3-D coordinates in the world
- (x, y): 2-D coordinates in the image
- $\mathbf{t} = (t_x, t_y, t_z)$: translational component of motion
- $\omega = (\omega_x, \omega_y, \omega_z)$: rotational component of motion

- $(u, v) = (\dot{x}, \dot{y})$: optical flow field

Let us start by deriving the equations relating motion in the 3-D world to the resulting optical flow field on the 2-D image plane. For simplicity we will focus on a single point in the scene $\mathbf{X} = (X, Y, Z)$.

Assume that the camera moves with translational velocity $\mathbf{t} = (t_x, t_x, t_z)$ and angular velocity $\omega = (\omega_x, \omega_y, \omega_z)$. Eq.(4.1) is used to characterize the movement of $\mathbf{X}$,

$$\dot{\mathbf{X}} = -\mathbf{t} - \omega \wedge \mathbf{X}, \tag{4.1}$$

which can be written out in coordinates as Eq.(4.2):

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = - \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} - \begin{bmatrix} \omega_y z - \omega_z y \\ \omega_z x - \omega_x z \\ \omega_x y - \omega_y x \end{bmatrix}. \tag{4.2}$$

Assume the image plane lies at $f = 1$, then $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$. Taking the derivative, we have

$$\dot{x} = \frac{\dot{X}Z - \dot{Z}X}{Z^2}, \dot{y} = \frac{\dot{Y}Z - \dot{Z}Y}{Z^2}. \tag{4.3}$$

Substitute $\dot{X}, \dot{Y}, \dot{Z}$ in Eq.(4.3) using Eq.(4.2), plug in $x = \frac{X}{Z}, y = \frac{Y}{Z}$, and simplify it, we get

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{4.4}$$

We can use these equations to solve the forward (graphics) problem of determining the movement in the image given the movement in the world. If we assume that the parameters $\mathbf{t}$, ω are the same for all the points, that is equivalent to a rigidity assumption. It is obviously true if only the camera moves. Else if we have independently moving objects, then we have to consider each object separately.

Can all the unknowns be recovered, given enough points at which the optical flow is known? There is a scaling ambiguity about which we can do nothing. Consider a surface S_2 that is a dilation of the surface S_1 by a a factor of k, i.e. suppose that the corresponding point of surface S_2 is at depth $kZ(x, y)$. Furthermore suppose that the translational motion is k times faster. It is clear that the optical flow would be exactly the same for the two surfaces. Intuitively, farther objects moving faster generate the same optical flow as nearby objects moving slower. This is very convenient for generating special effects in Hollywood movies!

4.3 Pure translation

If the motion of the camera is purely translational, the terms due to rotation in Eq. (4.4) can be dropped and the flow field becomes

$$u(x,y) = \frac{-t_x + xt_z}{Z(x,y)}, v(x,y) = \frac{-t_y + yt_z}{Z(x,y)}. \tag{4.5}$$

We can gain intuition by considering the even more special case of translation along the optical axis, i.e. $t_z \neq 0, t_x = 0, t_y = 0$, the flow field in Eq.(4.5) becomes

$$u(x,y) = \frac{xt_z}{Z(x,y)}, v(x,y) = \frac{yt_z}{Z(x,y)}; \tag{4.6}$$

or equivalently

$$[u,v]^T(x,y) = \frac{t_z}{Z}[x,y]^T \tag{4.7}$$

This flow field has a very simple structure, as shown in Figure 4.2. It is zero at the origin, and at any other point, the optical flow vector points radially outward from the origin. We say that the origin is the **Focus of Expansion** of the flow field. The proportionality factor $\frac{t_z}{Z}$ is significant because it is the reciprocal of the **time to collision** $\frac{Z}{t_z}$ There is considerable evidence that this variable is used by flies, birds, humans etc as a cue for controlling locomotion. Note that while we are unable to estimate either the true speed (t_z) or the distance to the obstacle (Z), we are able to estimate what truly matters for controlling locomotion. Sometimes nature is kind!

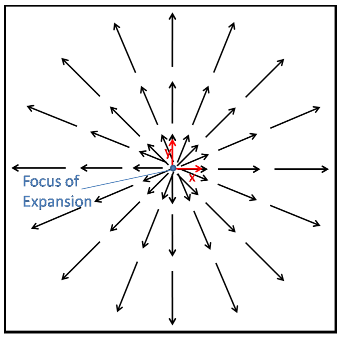


Figure 4.2: Optical flow field of an observer moving along the z-axis towards a frontoparallel wall

The case of general translation is essentially the same. We define the **Focus of Expansion** (FOE) of the optical flow field to be the point, where the optical flow is zero. Set $(u,v) = (0,0)$ in Eq.(4.5), we can solve for the coordinates of the FOE,

$$(x_{FOE}, y_{FOE}) = (\frac{t_x}{t_z}, \frac{t_y}{t_z}). \tag{4.8}$$

Note that the coordinates of the FOE tell us the direction of motion (we can't hope to know the speed, anyway!). It is also worth remarking that the FOE is just the vanishing point of the direction of translation.

Suppose we change the origin to the FOE by applying the following coordinate change to Eq.(4.5),

$$x' = x - \frac{t_x}{t_z}, y' = y - \frac{t_y}{t_z}, \tag{4.9}$$

then the optical flow field becomes

$$[u, v]^T(x', y') = \frac{t_z}{Z}[x', y']^T. \tag{4.10}$$

which should look very familiar. Thus the general case too corresponds to optical flow vectors pointing outwards from the FOE, justifying the choice of the term. Figure 4.3 shows such an optical vector field.

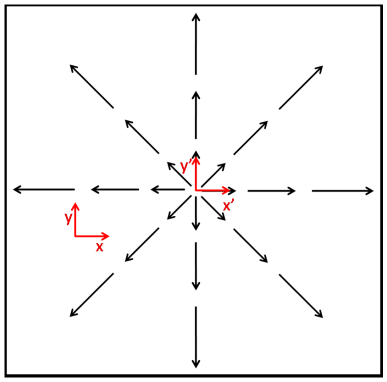


Figure 4.3: Optical flow vector field for general translational motion

We can also detect depth discontinuities from the optical flow field. If there is a sharp change in the lengths of flow vectors of two neighboring points, that indicates a discontinuity in depth. The ratio of their lengths tells us the ratio of their depths $(\frac{Z_1}{Z_2})$; however, we can't deduce the absolute depths (Z_1, Z_2), which is illustrated in Figure 4.4.

Thus optical flow is one of the most important cues to image segmentation (video segmentation, actually!). Even camouflaged animals (and snipers) must learn to stay very still to avoid detection.

4.4 General Motion

We begin by studying pure rotation. The most important thing to note is that the rotational component, obtaining by setting $\mathbf{t}$ to zero, has no dependence on Z. Therefor it conveys no information about the scene depth, only about the rotation of the observer. For moving animals in a stationary scene, this

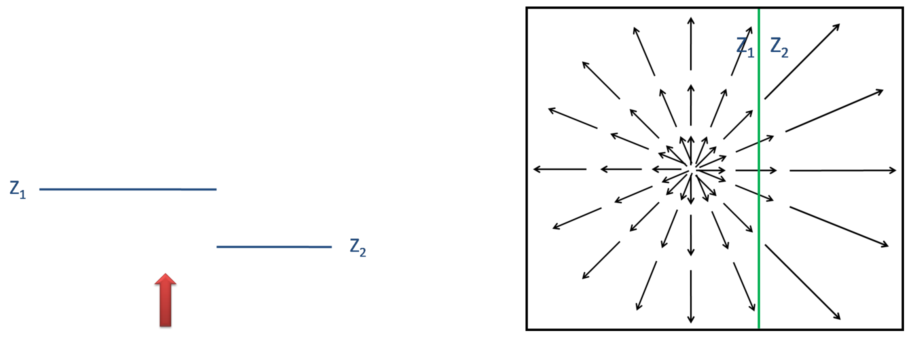


Figure 4.4: Depth discontinuity in optical flow field

commonly arises due to eye movements, which correspond to a rotation about the center of projection.

Thus the optical flow field corresponding to a general motion can be thought of as having a translational component very useful for inferring time to collision, depth boundaries in the scene, etc., and a rotational component which carries no information about the external 3D world. In the context of a moving animal where the rotational component is due to eye movements, some part of the animal brain has access to the rotational signal, since the eye movement was commanded by the brain itself. Hence the so-called **efference copy** carries information that can be used to subtract the rotational component. The residual is a purely translational flow field which can be be analyzed more straightforwardly. Amazingly, this is actually the case in humans (and probably in other animals with eye movements).

4.5 Summary

- Optical flow is the motion of the 3-D world projected on to the 2-D image. It can be used to derive cues about the structure of the 3-D scene as well as egomotion.

- The optical flow field for pure translation enables us to infer

  - The direction of movement, but not the absolute speed
  - The time to collision
  - Locations of depth discontinuities

4.6 Exercises

1. Implement the equations which relate the point wise optical flow to the six parameters of rigid body translation and rotation, and depth. Construct displays for some interesting cases.

2. As a test for the code that you have written in the previous exercise, suppose that I am driving my car along a straight stretch of freeway at a speed of 25 m/s. My eye height above the surface of the road is 1.25 m. What is the flow vector (in degrees/s)

   (a) At a point on the ground 25 m straight ahead.

   (b) At a point on the ground to my left at a distance of 25 m.

   (c) At points on the rear end of a 2 m wide car at a height of 1.25 m above the ground. This car has a headway of 25 m in front of me and is travelling at a speed of 20 m/s.

Chapter 5

Binocular Perspective

5.1 Stereopsis

Binocular Stereopsis (from stereo meaning solid, and opsis meaning vision or sight) is the visual cue to perceive depth from slightly different images produced by the two eyes. The slight difference in the images produced is called disparity.

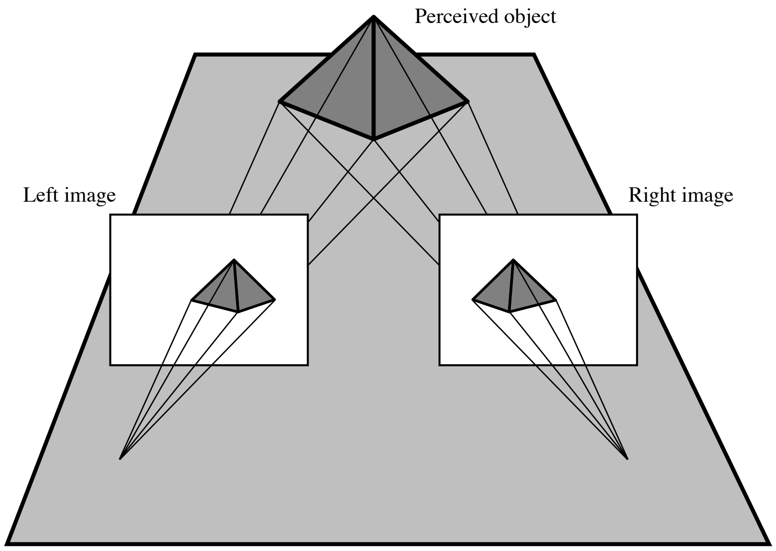


Figure 5.1: Binocular Viewing Geometry

Consider two eyes fixated at a point P, and consider the image plane coordinates of a point Q in the two eyes. Disparity is defined by the vector $(x^q_R - x^q_L, y^q_R - y^q_L)$. The first component is called the horizontal disparity and the second component is called the vertical disparity. By construction, the disparity of the fixation point, P is (0,0).

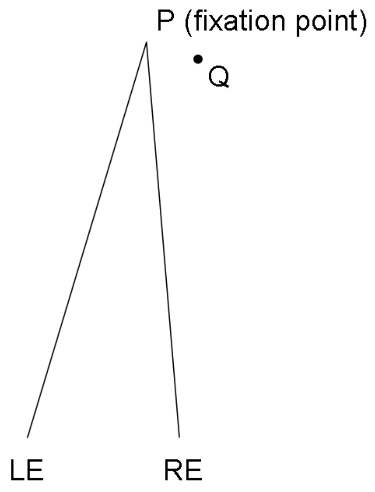


Figure 5.2: The eyes are fixated at P. Where does Q project?

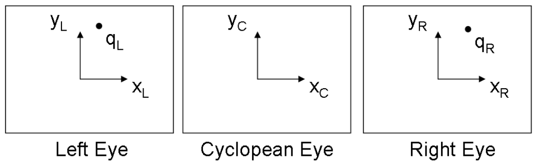


Figure 5.3: Defining disparity

5.2 A simple formula

Consider the simple case of eyes fixating at a point straight ahead. The disparity in this case can be easily computed from the figure below.

$$\tan\theta = \frac{b/2}{Z}$$

b is the baseline distance or the interocular distance

$$\tan(\theta - \delta\theta) = \frac{b/2}{Z + \delta Z}$$

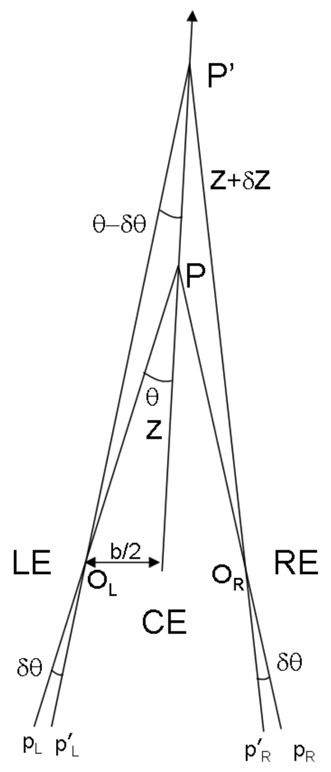


Figure 5.4: Pure vergence

For small angles $\tan\theta \approx \theta$

$$\begin{aligned}
\theta &= \frac{b/2}{Z} \\
(\theta - \delta\theta) &= \frac{b/2}{Z+\delta Z} \\
\delta\theta &= \frac{b/2}{Z} - \frac{b/2}{Z+\delta Z} \\
&\approx \frac{b\delta Z}{2Z^2}
\end{aligned}$$

Since actual disparity is $2\delta\theta$

$$\text{disparity} = \frac{b\delta Z}{Z^2}$$

The depth difference δZ humans can discriminate can now be computed using the above formula. The average intraocular distance for humans is 6cm. It is also known that human eyes are capable of measuring an angle disparity $\delta\theta$ of $5''$. Using these values the δZ humans can discriminate at a distance of 1 m can be computed to be 0.4 mm. This improves by 10 times to 0.04 mm at reading distance. On the other hand, at very large distances of say 1km, it deteriorates to 400m. Hence, it is very good at close distances but is not that well suited for depth cues at large distances.

5.3 Correspondence Problem

The eyes of a normal human fixate – both eyes foveate at the same point. In a 2 camera system, there is no such constraint. The common case between the two is when the two camera system has parallel optical axes converged at infinity.

Also, we don't need to be constrained to just two cameras. In computer vision, we consider multiple cameras in general. So, the question is, what is the right counterpart to optical flow and disparity in the multiple camera case? We might consider pairwise differences, but this is an inelegant solution that adds unnecessary comparisons. Instead we relate the coordinate system of each camera to that of the world

Figure 5.5 shows a 3 camera situation, where p_1, p_2, and p_3 are corresponding points, which all correspond to the same point P out in the world. The correspondence problem is that of identifying for a point p_i in one image its partner p_j in another image.

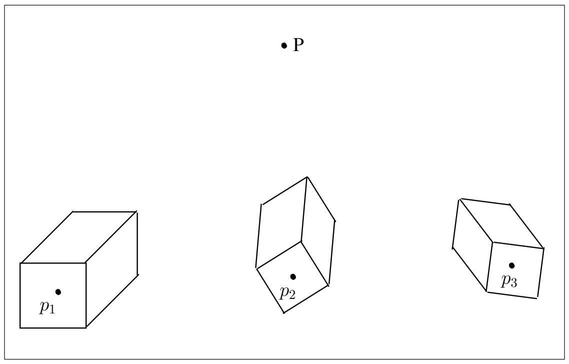


Figure 5.5: A 3 camera example of the correspondence problem.

5.4 Epipolar Geometry

Consider the two camera situation of Figure 5.6, where the left camera has center of projection C_1, the right camera has a center of projection C_2, and M is any 3-D point, not necessarily a fixation point. The points M, C_1, and C_2 form a plane that is defined to be the epipolar plane through M. The intersection of the epipolar plane with an image plane forms a line, called the epipolar line. For all points on the epipolar plane, points on the epipolar line of the left camera must have partners on the epipolar line of the right camera. In the figure, points m_1 and m_2 on the epipolar lines of the two cameras both correspond to the point M. This means we only have to do a one-dimensional search to find corresponding points!

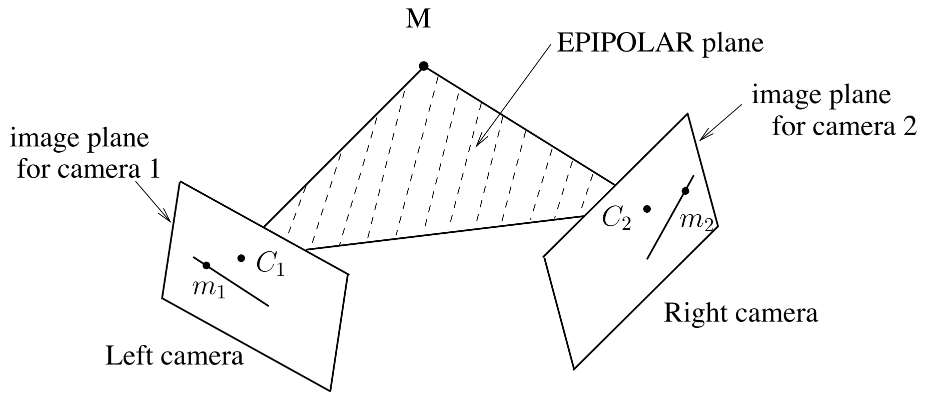


Figure 5.6: Illustration of epipolar geometry for two cameras.

5.5 Measuring optical flow and disparity

The measurement of optical flow requires finding for a point in one time frame t_1 its corresponding point in another time frame t_2. The measurement of disparity requires solving the correspondence problem between two images which are acquired at the same time, but from two different cameras.

Typically, some kind of block matching strategy is used. Consider a block of pixels around pixel p_1 in the first image, we compare it to the block of pixels around a pixel p_2 in the second image. We can compare these blocks of pixels by regarding them as vectors and just finding the Euclidean distance between these vectors e.g. if we consider 3×3 blocks we will be considering 9 dimensional vectors. A smaller distance means that it is more likely that the pixels correspond. In addition, for stereopsis, we use the epipolar constraint to reduce the search to corresponding epipolar lines.

Chapter 6

How surfaces project

Consider the following figure:
We observe that the lines parallel to the observer's sight axis converge to a vanishing point. We also note a distance effect; that is, tiles that are further away appear smaller. Even though the discs are circular, they appear elliptical and increasingly oblong as the distance increases.

In more concrete terms, we have two effects from perspective projection:

1. *Distance Scaling*: The object is scaled by $\frac{1}{r}$, where r is the distance.
2. *Foreshortening*: The aspect ratio of the object changes (in our case, the square tiles and circular discs).

   To easily illustrate this effect, draw a circle on a sheet of paper. Hold the paper parallel to your (the viewer's) xy plane, and then gradually rotate the paper about the x axis. As the paper rotates, the circle becomes an ellipse.

6.0.1 Slant and Tilt

In the above illustrative example, the aspect ratio of the ellipse may be found according to $\cos\sigma$, where σ is the *slant angle.* That is, the slant σ is the angle between the ray from the viewer and the observed object's surface normal.

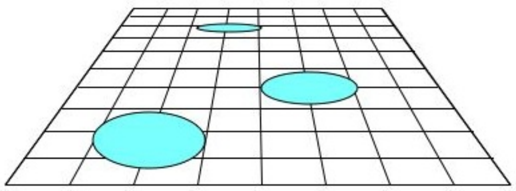


Figure 6.1: Image exhibiting depth cues.

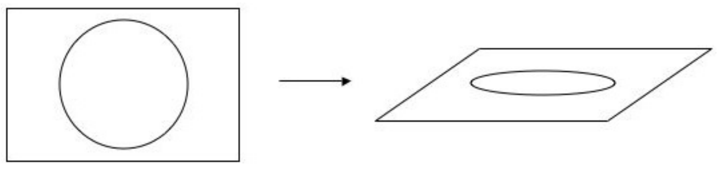


Figure 6.2: Simple example to illustrate foreshortening.

We might also say that the slant is the amount by which the surface differs from the local frontoparallel direction.

- For $\sigma = 0$, the resulting aspect ratio is 1, and the viewer observes a circle.
- For $\sigma = 90^\circ$, the aspect ratio is 0 or ∞, and the viewer observes a line.

Since the surface normal is a 3D unit vector, it has 2 DOF. Physically, we interpret this by saying that by specifying σ, the first of the DOF's, we obtain a cone associated with our value of σ. We still need to assign the other DOF, which is *tilt*. Tilt is defined as the direction in which the surfaces moves away, measured clockwise from the x-axis. We may also define tilt as the gradient of r (the direction in which r changes most rapidly). Once both slant and tilt have been specified, we may obtain the surface normal.

Chapter 7

Pose Estimation

7.0.2 Introduction

Pose estimation is the problem of starting from an object's projection in an image, and determining how it is oriented in the world.

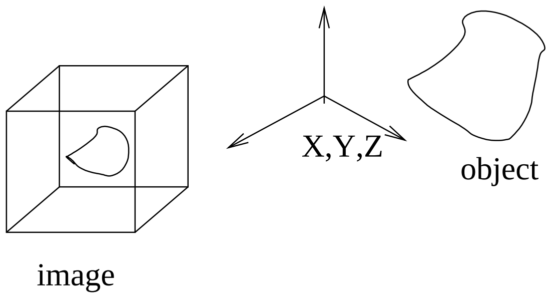


Figure 7.1: Pose estimation maps an image to an object

7.0.3 Perspective projection Review

Consider a point (X, Y, Z) in the world, where measurements are made with respect to a coordinate system with origin at the center of projection and optical axis being the Z axis. Its projection in the image plane (x, y) is given by the following equations:

- $x = \dfrac{fX}{Z}$
- $y = \dfrac{fY}{Z}$

7.0.4 Scaled Orthography

Consider imaging an object where the depth variation in the object dZ is small compared to its distance from the camera Z. Therefore we can regard

Z as approximately constant for points on the object, and since f is a constant anyway, we get the following simplified equations relating image plane coordinates to scene coordinates. This assumption is called scaled orthography.

- $x = \alpha X$
- $y = \alpha Y$

where α is the constant Z/f. Note that this simplification implies that there is no vanishing point for parallel lines.

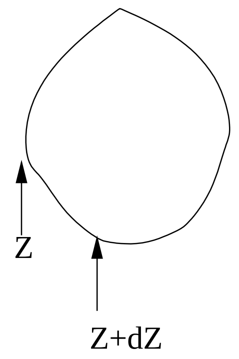


Figure 7.2: The value dZ is very small compared to the distance Z

7.1 Pose estimation for rigid objects

If we know 3 points of the model and their corresponding projection onto the image, we can solve for the pose parameters. The pose parameters consist of the six degrees of freedom for a rigid object, 3 for rotation and 3 for translation. Under scaled orthography, the centroid (mean of the points) of the object projects to the centroid of the image points, translation is straight forward. This simplification leaves us with 4 unknowns, rotation (3 parameters) and scale (1 parameter).

7.1.1 Review: Affine Transformation

We can estimate affine transform in 2D given 3 points and how they transform

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

With the three pairs of corresponding points, 6 linear equations can be extracted to solve the 6 unknown values in matrices A and T unless the points are colinear (degenerate case).

$$\begin{tikzcd}
\begin{array}{l}\text{Image } R^2\\ l_1\ l_2\end{array} \arrow[r, "L"] \arrow[d, "*"] & \begin{array}{l}\text{Image } R^2\\ l_1'\ l_2'\end{array} \\
\begin{array}{l}\text{World } R^3\\ l_1, 0\hat{z}\\ l_2, 0\hat{z}\end{array} \arrow[r, "U"'] & \begin{array}{l}\text{World } R^3\\ v_1 = l_1' + c_1\hat{z}\\ v_2 = l_2' + c_2\hat{z}\end{array} \arrow[u, "\Pi"']
\end{tikzcd}$$

Figure 7.3: Transformation relationships from Huttenlocher and Ullman's paper

7.1.2 Transformation

Figure 7.3 relates the transformation in the linear image plane to the corresponding transformation in the model plane, formed by the three chosen points. v_1 and v_2 represent orthonormal vectors in this model plane.

- L is the linear transformation in the image space.
- U is the scaling-rotation-translation transformation in the model space
- $*$ lifts the coordinates from the image space to model space with $Z = 0$
- $\prod$ moves model points into 2D by dropping Z

When a set of points undergo a 3D rigid transformation in the scene, their projection in the image is not a 2D rigid transformation but an affine transformation. To map from the image plane to the projection plane, we need to solve for c_1 and c_2 and then for the scaling, rotation, and translation. We use the fact v_1 and v_2 are orthonomal (meaning $\|v_1\| = \|v_2\|$ and $\|v_1\|^2 = \|v_2\|^2$)to produce two equations to solve for the C unknowns (reference Huttenlocher and Ullman, page 200). There will be two possible solutions, which are reflections of each other.

7.2 Taylor's algorithm - reconstructing a stick figure

7.2.1 Degrees of Freedom

In a stick figure, like in figure 7.4, each limb has 4 degrees of freedom, the torso has 8, and there is an unknown scale factor s. We have 24 independent measurements from the (u, v) coordinates of the 12 points in the image; thus we have 1 more unknown than equations. Our solution therefore will be a one-parameter family, rather than a unique reconstruction.

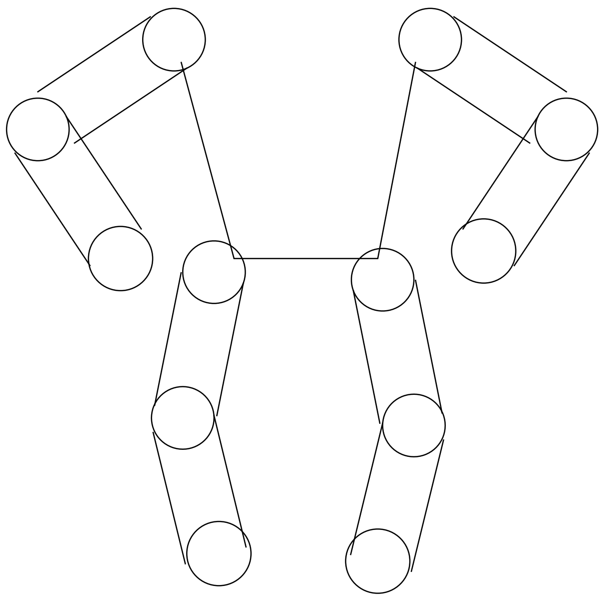


Figure 7.4: generalized pose

7.2.2 Scaled Orthographic Projection

Taylor assumes a scaled orthographic projection model:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

This projection shrinks X and Y while dropping Z.

7.2.3 Key Equation

Taylor's method builds on foreshortening, the fact the length of a segment in an image depends on its angle with the front parallel plane. We proceed by taking two end points of a segment and their projections onto the image to get the following equations:

$$(U_1 - U_2) = s(X_1 - X_2)$$

$$(V_1 - V_2) = s(Y_1 - Y_2)$$

$$l^2 = (X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2$$

$$dZ = \sqrt{l^2 - \frac{(U_1 - U_2)^2 + (V_1 - V_2)^2}{s^2}}$$

With these equations, the depth can be calculated if s is known. We can place bounds on s because the quantity under the square root can not be negative. When we track humans in video we have other constraints. Both dZ and s can only change slowly from frame to frame.

Chapter 8

Radiometry of Image Formation

8.1 Radiance and Irradiance

Radiance is the power traveling at some point in a specified direction per unit foreshortened area per unit solid angle.

The units of radiance are $Wm^{-2}sr^{-1}$, where sr is a steradian, the unit of solid angle. We will show later that radiance along a ray does not diminish (assuming no absorption of energy by the medium).

The power per unit area leaving a point is obtained by integrating over the solid angle

$$\int_{\omega} L\cos\theta d\omega$$

where

$$d\omega = \sin\theta d\theta d\phi$$

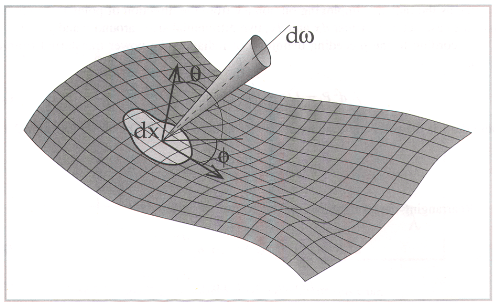


Figure 8.1:

In these equations, $d\omega$ is a differential solid angle, θ is the angle between the outgoing direction and the normal to the surface, and ϕ is the other angular coordinate (in spherical coordinates) of this outgoing direction (defined by projecting it to the surface and measuring the angle with respect to a a reference direction.) This is illustrated in Figure 8.1.

Irradiance or Radiosity is the total power per unit area coming into or leaving a location on the surface.

Irradiance can be measured by integrating the radiance over all directions. Thus, the units are Wm^{-2}.

8.2 BRDF

Definition 5 *BRDF, or the Bidirectional reflectance distribution function, is the amount of outgoing radiance / incoming radiance. By convention,* θ_O, ϕ_O *are for outgoing, and* θ_i, ϕ_i *are incoming.*

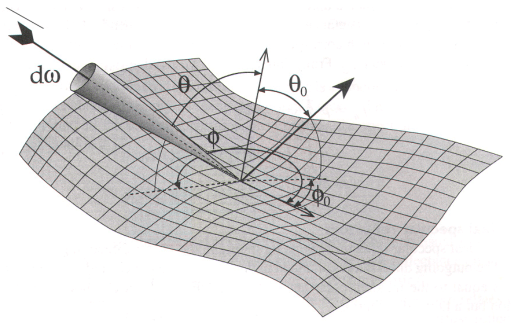


Figure 8.2:

8.3 Shape from Shading

A camera measures incoming radiance in the direction of the camera. (See figure 8.3) Knowing the light source and the measurements of irradiance, we can find out characteristics of the surface.

Let:
$\hat{s}$: Direction of light source
$\hat{n}$: Lambertion surface normal
ρ : albedo
The irradiance at the camera is:

$$E(x, y) = \rho \; \hat{n}(x, y) \cdot \hat{s}$$

Note: there are two variables and one equation, this can't be solved locally.

This is the *shape from shading problem*: how to determine surface shape from the the difference in measured irradiance (See figure 8.4).

In this case, it is about figuring out $\hat{n}(x, y)$ from the measured irradiance at the camera.

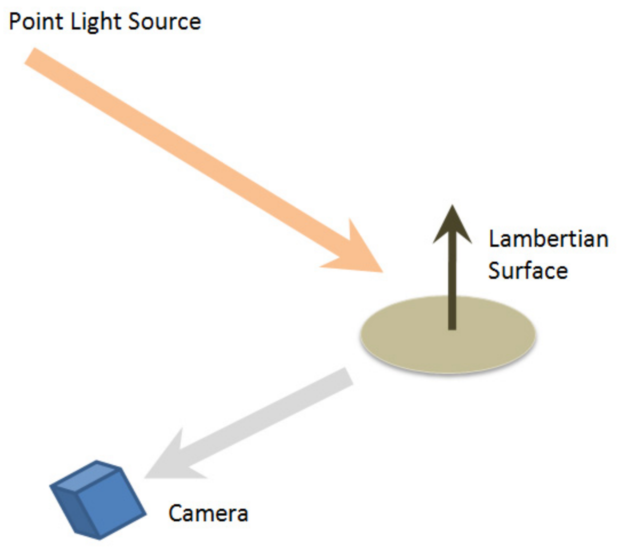


Figure 8.3: The irradiance computation at a camera with a point light source and a Lambertian surface

A surface can be described as:

$$Z - f(x, y) = 0$$

For this surface, the surface normal can be expressed as:

$$\hat{n} = \frac{[-f_x; -f_y; 1]}{\sqrt(1 + f_x^2 + f_y^2)}$$

The light source is infinitely far away:

$$\hat{s} = [s_x; s_y; s_z]$$

Then radiance in image plane coordinate (x, y):

$$E(x, y) = \rho(\frac{-f_x s_x - f_y s_y - s_z}{\sqrt(1 + f_x^2 + f_y^2)})$$

Note that surface is:

$$Z = f(x, y)$$
$$f_x = Z_x$$
$$f_y = Z_y$$

$$E(x, y) = \rho(\frac{-s_x Z_x(x, y) - s_y Z_y(x, y) - s_z}{\sqrt(1 + f_x^2 + f_y^2)})$$

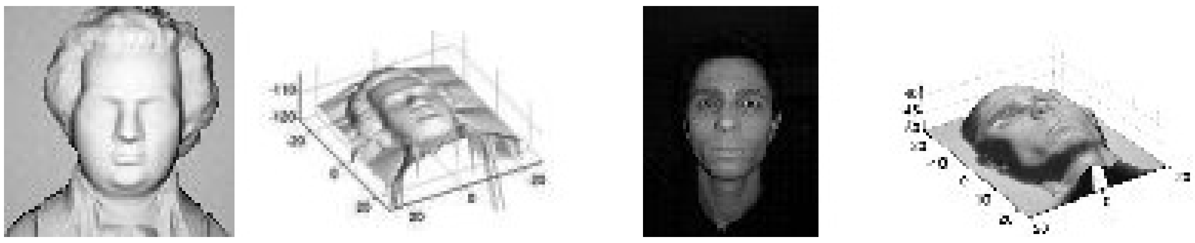


Figure 8.4: Shape from shading problem (credit: Prados et al, INRIA Research Report RR-5297)

Here, the unknown function is $Z(x, y)$.

We can compute partial derivatives:

$$\frac{\partial Z}{\partial x} = Z_x(x, y)$$

$$\frac{\partial Z}{\partial y} = Z_y(x, y)$$

It was solved by Horn in 1970 using characteristic strips.

8.4 Deviations from the model

The radiance formula we produced is: $E(x, y) = \rho\, \hat{n}(x, y) \cdot \hat{s}$

A realistic scenario may deviate from this model in three ways:

1. Variations in reflectance
2. Variations in surface geometry
3. Deviation from a collimated light field

Variations in reflectance: ρ may not be a constant w.r.t. (x, y). For example, the surface may be a piece of paper with printed characters. The part with dark ink would have $\rho = 0$, and the white background of the printed page would have $\rho = 1$.

In addition, specularity on the surface and other deviations from the Lambertian assumption will deviate from this

Variations in surface geometry: Mesostructure (or relief textures) on the surface affect the scattering of light on the surface. These bumps on the surface may cause reflection to be dominated by self shadowing and self occlusion.

Deviation from a collimated light beam: "Light field" is not just a collimated light beam, there could be:

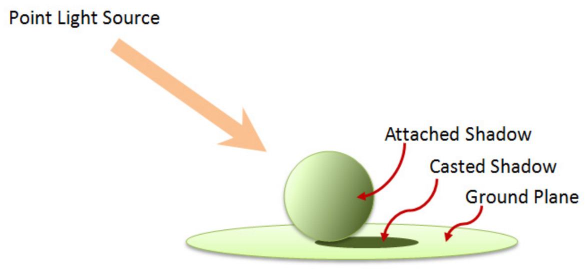


Figure 8.5: A ball on the ground plane, with point light source, which help illustrates an attached shadows and a casted shadow

1. Attached shadows (See figure 8.5)
2. Cast shadows
3. Mutual Illumination ("Reflexes")

8.5 Higher-level cognitive processing in human vision

8.5.1 Adelson's checkerboard

Figure 8.6 (left) (*Adelson's checkerboard*) demonstrates a seemingly paradoxical situation. We see a checkerboard, of which squares A and B should correspond to alternately shaded "black" and "white" squares. However, is A really darker than B? In fact, the two squares have the same pixel brightness in this image, which can be verified upon closer inspection or by modifying the context as in Figure 8.6 (right).

Recall that the brightness recorded by a camera corresponds to the outgoing radiance in the direction of the camera. This radiance depends on the surface's reflectance properties: in this case, the BRDF value of ρ is greater for square B because it is a "white" square. But the radiance also depends on other factors such as foreshortening effects, shadows, and mutual illumination. In particular, the shadow of the large cylinder causes the light field at square B to be less in this direction; when integrated over all directions, the incoming irradiance at square B is less than at square A.

An image, recorded as brightness values, is the product of a material's reflectance properties and its light field. But we are not interested in this pixel brightness, so the brain must therefore *unmultiply* with higher levels cognitive processing in order to determine ρ, which corresponds to what we are ultimately interested in determining: information about an object. In

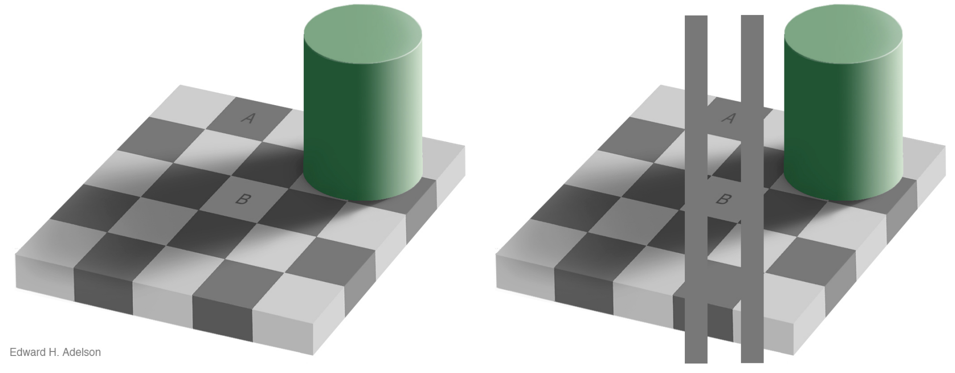


Figure 8.6: Adelson's checkerboard. On the left, the squares A and B are interpreted as being different, even though they actually have the same pixel brightness (outgoing radiance). The figure on the right is provided as more evident proof.

other words, given a visual field (2-dimensional retinal image) we are able to infer a visual world (3-dimensional external reality). For example, when we see a texture gradient we infer that surface's 3D geometry (e.g., wooden slats on the right-side wall of Soda 306).

8.6 Specular reflection and physical color

Specular reflection occurs when a light ray traveling through air hits a material surface. For a perfectly specular surface, such as a mirror, we characterize this as *Fresnel* reflection for which intuitive Laws of Reflection apply: the incidence angle between the incoming ray and the surface normal must equal the reflectance angle between the outgoing ray and the surface normal; and all three rays must be coplanar.

The physical reality for most surfaces, however, is that there is a diffuse component in addition to the Fresnel reflection. Some proportion of the incoming rays will be scattered within the material and reflected back out in various directions. Typical surfaces therefore combine reflection and diffusion to various degrees. Surfaces which reflect much of the incoming light are sometimes called *glossy* while those that diffuse it are considered *matt*. While there is no significant effect of color on the reflected component, the diffuse component often depends on the incoming light's wavelength. Thus glossy surfaces have *highlights* corresponding to the color of the incoming light, whereas matt surfaces tend to express the material's characteristic color.

In physical terms, we can express the effect of color as a dependence of the light field or BRDF on the wavelength λ of the incoming light. We can denote the BRDF as:

$$\rho(x, \theta_o, \phi_o, \theta_i, \phi_i, \lambda)$$

where x is a 3D position, and the other parameters denote the incoming and outgoing directions of (ir)radiance. Similarly, we can denote the light field as:

$$L(x, \theta, \phi, \lambda)$$

A material will exhibit higher values of the reflecance function ρ for those wavelengths λ which are characteristic of its color. A typical light field, from a source such as sunlight, will have radiance over a continuous spectrum of many wavelengths.

8.7 Psychophysical interpretation of color

Color perception is a psychophysical concept. The physical characteristics have been described as a dependence of the light field or BRDF on an incoming ray's wavelength. Visible light is a form of electromagnetic radiation, which may be characterized as combination of infinitely many different wavelengths. Despite a physicist's account of infinitely many colors, artists have long believed that there is a basic set of three pigments from which all other colors can be mixed; indeed, they were correct in describing the way humans perceive color.

Experiments to demonstrate the three-dimensional basis of human color perception were performed by Young in 1804, Maxwell in the 1850's, and Helmholz in the 1860's. In modern terms, they determined that there were three kinds of photoreceptors (cones) in the human eye, each absorbing light in differing ways. For each of the cone types would be an associated probability distribution for the absorption of a photon of a given wavelength, as shown Figure 8.7 for cones reactive to to (S)hort, (M)edium, and (Long) wavelengths in the range 400-700 nm. If the absorptions distribution for a cone of type i is related as $R_i(\lambda)$, then the response of a cone of type i is integrated over the range of wavelengths:

$$\int L(\lambda)R(\lambda)d\lambda$$

where $L(\lambda)$ is the number of photons of wavelength λ that are received at the cone.

Note that human color perception is a projection from an infinite-dimensional spectrum of wavelengths down to a three-dimensional response of the cone types. The phenomenon of *metamerism* occurs when we have two different lights $L_A(\lambda)$ and $L_B(\lambda)$ such that the cone response is the same for the three types. In the natural world, however, light spectra tend to be smooth enough that metamerism is not a major issue.

Some other remarks about human color vision:

- color blindness (such as red-green color blindness) occurs when cones of a certain type are not present in the retina. It is more common for

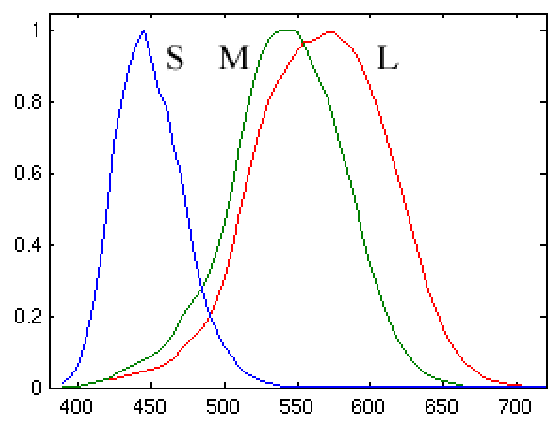


Figure 8.7: Normalized response spectra of human cones to monochromatic light at various wavelengths (nanometers).

males because the genes needed for proper color perception are found on the X chromosome.

- Humans have three cone types perhaps as an evolutionary tradeoff. Many animals have no color perception, while some can have up to ten cone types. Having more cone types means that there is less detailed resolution, however, since the density of cones on the retina is limited.

Chapter 9

Low-level Vision

9.1 Linear image processing operations

We have so far studied the process of image formation, considering the geometry and radiometry that produce a 2-dimensional image $I(x, y)$ – or its color components I_R, I_G, I_B – where this irradiance is proportional to radiance from the external scene in the direction of the camera. The image is therefore a 2-dimensional signal; an additional temporal dimension can be present for video signals. The decoding problem with which we are concerned is to infer the external world from these image signals. First, we must become acquainted with basic signal processing concepts.

Consider a system S that takes an input signal $v(t)$ and outputs $w(t)$, as in Figure 9.1. This system can have the following properties:

i Linearity:
If $v_1 \mapsto w_1$ and $v_2 \mapsto w_2$ then $\alpha v_1 + \beta v_2 \mapsto \alpha w_1 + \beta w_2$.

ii Shift invariance:
If $v(t) \mapsto w(t)$ then $v(t - \tau) \mapsto w(t - \tau)$.

iii Characterization of linear shift-invariant systems:
Any linear shift-invariant system S can be fully characterized by an

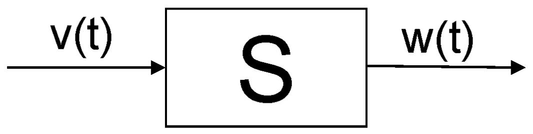


Figure 9.1: Block diagram of a system S that takes an input signal $v(t)$ and produces an output signal $w(t)$.

associated function h called its *impulse response.* In particular, $w = v * h$.

For this final point we must explain the notion of *convolution*, an operation with various uses in image processing such as blurring or edge detection. Mathematically, the convolution operation denoted by $*$ is as follows:

$$f * g(t) = \int_{-\infty}^{\infty} f(u)g(t-u)du$$

Graphically, the operation is depicted in Figure 9.2 as the area under the curve $f(u)g(t-u)$. Electrical engineers may be familiar with the intuition that convolution is a “flip and drag” operation. Note that convolution is linear, distributive, commutative, and associative:

$$\begin{gathered}(\alpha f_1 + \beta_2 f_2) * g = \alpha f_1 * g + \beta_2 f_2 * g \\ f * g = g * f \\ f * (g * h) = (f * g) * h\end{gathered}$$

9.2 Gaussian Functions

Following three functions are core functions in image processing and they are very important. G_σ is called Gaussian function.

$$G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$\begin{aligned}G'_\sigma(x) &= \frac{d}{dx} G_\sigma(x) = -\frac{1}{\sigma}\left(\frac{x}{\sigma}\right) G_\sigma(x) \\ G''_\sigma(x) &= \frac{d^2}{dx^2} G_\sigma(x) = \frac{1}{\sigma^2}\left(\frac{x^2}{\sigma^2} - 1\right) G_\sigma(x)\end{aligned}$$

Figure 9.3 shows how these three functions look like.

$G'_\sigma(x)$'s maxima/minima occur at $G''_\sigma(x)$'s zeros. And, we can see that $G'_\sigma(x)$ is an odd symmetric function and $G''_\sigma(x)$ is an even symmetric function.

9.3 Derivatives and Convolution

Taking a derivative is a linear operation. Since differentiation is linear, it can be performed by convolution with some function h. Using the commutativity and associativity properties of convolution, we can show that

$$(I * f)' = (I * f) * h = (I * h) * f = I' * f$$

By similar reasoning, we can show that $(I * f)' = I * f'$

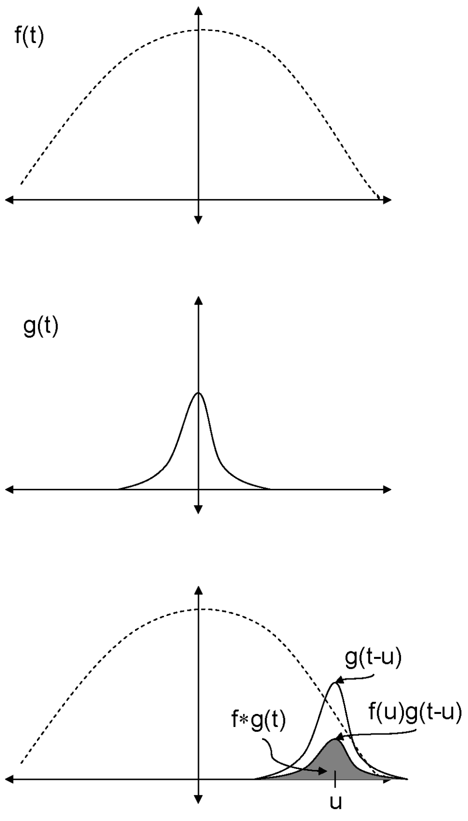


Figure 9.2: The convolution operation. The value of $f * g(t)$ is the area under the curve $f(u)g(t-u)$.

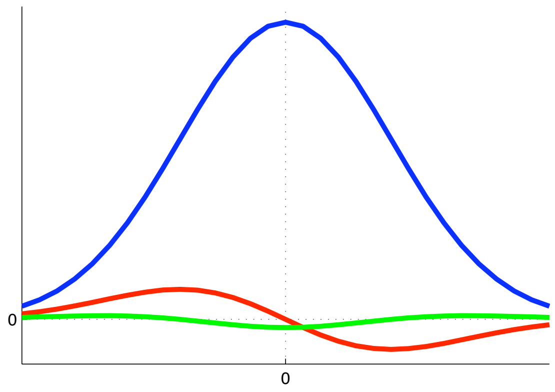


Figure 9.3: $G_\sigma, G'_\sigma, G''_\sigma$

9.4 1D Image Processing

Consider a one dimensional image, which is a scanline of a two dimensional image:

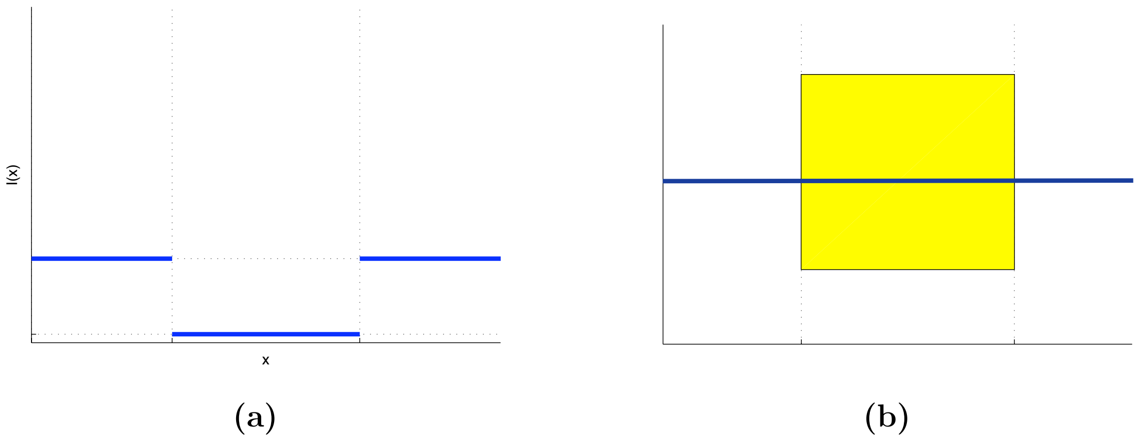


Figure 9.4: Image Function

Finding discontinuities in the image is important, because they correspond to boundaries of objects. We look for these by trying to find where $||I'(x)||$ is large.

Let

$$I(x) = \underset{\sim}{I}(x) + \epsilon \sin(\omega x)$$

where $\epsilon \sin(\omega x)$ is used to model noise. Then, what happens to the derivative?

$$I'(x) = \underset{\sim}{I}'(x) + \epsilon\omega \cos(\omega x)$$

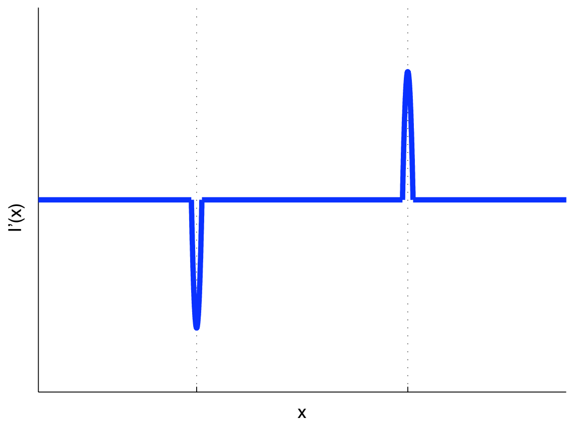


Figure 9.5: $I'(x)$

The response due to the noise has magnitude $\epsilon\omega$; the higher the frequency ω, the effect of noise is amplified in I'. We deal with this problem by 'smoothing' , by convolving with a Gaussian function.

9.5 Convolving with a Gaussian

Convolving with a Gaussin smooths the signal and thus "denoises" I. In the above example, if we choose σ to be larger than a period of the sine wave, then it will be a good choice. In fact, if the noise has a higher frequency, this is easier to kill; because we only have to choose smaller σ for the Gaussian

We can do both the smoothing and differentation in one pass.

Let f be $G_\sigma(x)$. Now, for any f

$$I' * f = (I * f)' = I * f'$$

Thus, all we need to do is to convolve the image $I(x)$ with $G'_\sigma(x)$.

9.6 Delta function

There are two kinds of delta "functions". The Dirac delta function is what we use in this class.

1. Kronecker delta function

$$\begin{aligned} \delta_{ij} &= 1 \; if\; i = j \\ &= 0 \; otherwise \end{aligned} \tag{9.1}$$

2. Dirac delta function

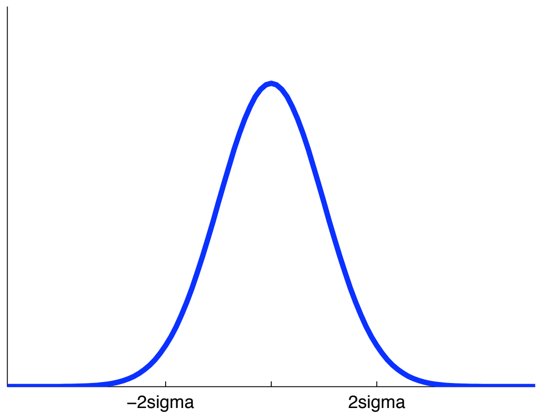


Figure 9.6: Gaussian Function

In the Gaussian function, Figure 9.6, as σ becomes smaller, area under the curve stays the same, but the function becomes more narrowly distributed.

$$\delta(x) = \lim_{\sigma \to 0} G_\sigma(x)$$

You use $\delta(x)$ as part of an integral also involving another function. The important property (called Sifting property):

$$\int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = f(x_0)$$

Aside Note

The BRDF of a perfectly specular material is modeled by a δ function.

$$\rho_{bd}(\theta_o, \phi_o, \theta_i, \phi_i) = \rho_s(\theta_i) \cdot 2\rho(\sin^2\theta_o - \sin^2\theta_i)\rho(\phi_o - \phi_i \pm \pi)$$

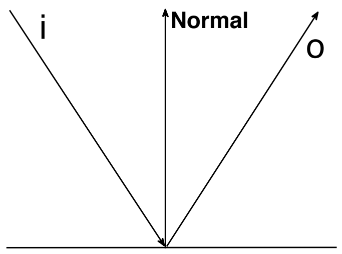


Figure 9.7: BRDF Example

Basically, the action is when $\theta_o = \theta_i$ and $\phi_o = \phi_i \pm \pi$. This corresponds to angle of incidence being equal to angle of reflection, and the incident ray, reflected ray and the normal all being the same plane.

9.7 Multi-dimensional Gaussian

Now, we will discuss Multi-dimensional Gaussian function. For example, in 2D:

$$\text{Anisotropic: } G_{\sigma_x,\sigma_y}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)} \tag{9.2}$$

$$\text{Isotropic: } G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}} \tag{9.3}$$

Note that isotropic function is just special case of anisotropic when we set $\sigma_x^2 = \sigma_y^2$ and $r^2 = x^2 + y^2$. And, also, by setting $r^2 = x^2 + y^2$, we get the function in polar coordinates, (r, θ). There is no dependence on θ as the function is isotropic.

We often use

$$f_1(x,y) = G'_{\sigma_1}(x) G_{\sigma_2}(y) \tag{9.4}$$

$$f_2(x,y) = G''_{\sigma_1}(x) G_{\sigma_2}(y) \tag{9.5}$$

We also consider rotated versions of these Gaussian derivative functions.

$$Rot_\theta f_1 = G'_{\sigma_1}(u) G_{\sigma_2}(v) \tag{9.6}$$

$$Rot_\theta f_2 = G''_{\sigma_1}(u) G_{\sigma_2}(v) \tag{9.7}$$

where we set

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

These are useful when we convolve with 2D images, e.g. to detect edges at different orientations.

HOMEWORK: Visualise all these functions in matlab and also convolve with other functions. This will give you the chance to see how they all work.

9.8 Overview

1 Fourier Transforms

2 Natural Image Statistics

3 Pyramids

9.9 Fourier Transforms

Fourier series are a tool used to analyze periodic functions by decomposing them into a weighted sum of sine or cosine waves. Fourier transforms are a generalization of Fourier series. They are used to approximate a non-periodic function.

Definition 6 *A Fourier Transform is defined as:*

$$F(s) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi xs}dx$$

$s \in R$ is a real value which is the counterpart of the finite set of frequency values used for Fourier series.

The Fourier transform takes a function $f(x)$ in the space domain and transforms it into a function $F(s)$ in the frequency domain. The inversion formula which transforms a function from the frequency domain into the space domain is:

$$f(x) = \int_{-\infty}^{\infty} F(s)e^{i2\pi xs}ds$$

A function such as $f(x) = cos(\pi x)$ in the space domain will have a very simple form in the Fourier domain:

$$\frac{1}{2}\delta(s+\frac{1}{2}) + \frac{1}{2}\delta(s-\frac{1}{2})$$

Definition 7 *In two dimensions,*

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i2\pi(xu+yv)}dxdy$$

In general, the Fourier transform is a complex-valued function, so it can be represented in terms of amplitude and phase.

Properties of Fourier Transforms:

- Linearity
  The Fourier transform in linear:
  $h(x) = \alpha f(x) + \beta g(x) \leftrightarrow H(s) = \alpha F(s) + \beta G(s)$ where the lower case functions are functions in the space domain and the corresponding upper case functions are their Fourier transforms.

- Convolution Theorem
  Convolution in the space domain is the same as multiplication in the Fourier domain:
  $h = f * g \leftrightarrow H(s) = F(s)G(s)$

- Separability
  $h(x,y) = f(x)g(y) \leftrightarrow H(u,v) = F(u)G(v)$

- Gaussian Property
  Gaussian in space domain $\leftrightarrow$ Gaussian in frequency domain

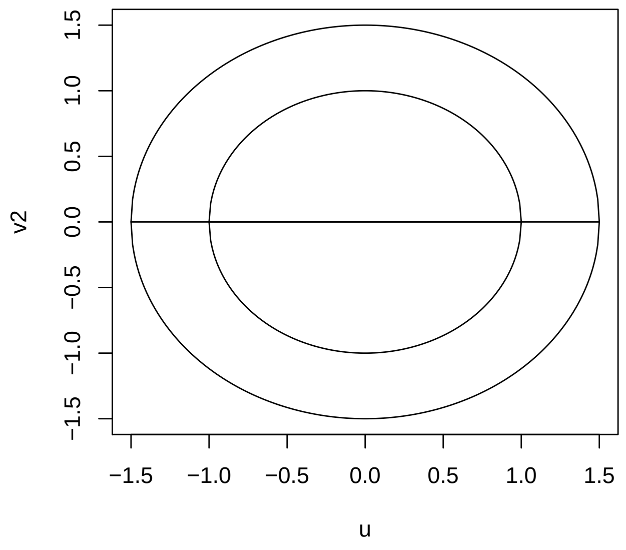


Figure 9.8: Disk in Fourier Domain(in between circles)

9.10 Natural Image Statistics

We can think of images as two dimensional functions $I(x, y)$. However, images are a special kind of functions that arise by taking pictures of the real world. So these functions have some special properties:

- Scale Invariance

  The structure of an image is the same at any scale. If we take an image, crop out a window and magnify it, crop out a window of the new image and magnify it, and so on, we get a sequence of images whose statistics are the same.

  The notion of self-similarity at multiple scales was introduced by Mandelbrot. He defined a fractal, a geometric shape that can be subdivided into parts, each of which is at least approximately a reduced-size copy of the whole.

  An interesting consequence of scale-invariance is that the Fourier power spectrum folows $\frac{A}{f^2}$. Consider a disc f to $f + df$ as shown in Figure 9.8.

  $$\int_{f_1}^{nf_1} 2\pi f g(f) df = K$$

  where f is the radial component, assuming angular symmetry, and $g(f)$ is the Fourier transform. In order for this property to hold, $g(f)$ has to be of type $\frac{c_1}{f}$. The power spectrum, defined as $||g(f)||^2$ is thus of type $\frac{c_2}{f^2}$.

- Power Law Distribution of Areas of Regions

  An alternative way to think about scale-invariance is by segmenting the image into regions corresponding to different objects. We can find the areas of the different regions obtained by segmentation and plot a histogram of region areas. The histogram, obtained by combining data from many images, shows that the number of regions of a certain area decreases with the region area.

- Heavy Tails in Distribution of Filter Outputs

  Suppose we convolve an image I with some filter f, for examples $f = [-1, 1]$. $R = I * f$. The histogram of R is peaked at 0, is symmetric, and has heavy tails. This is because most values of R in an image will be around 0, but there will be bigger or smaller values of R, corresponding to edges in the image. Low contrast edges will be more frequent than high contrast edges, which will be located towards the tails of the distribution.

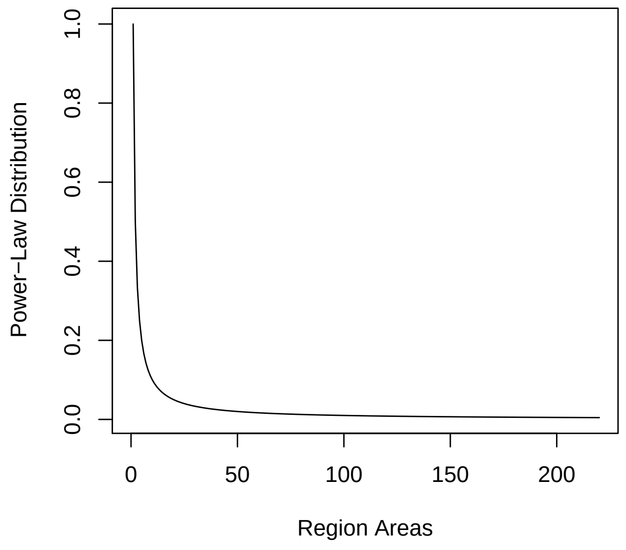
Definition 8 *The concept of heavy tails is expressed by the kurtosis of a distribution, defined as:*
$\frac{E(x-\mu)^4}{\{E(x-\mu)^2\}^2} - 3$
The kurtosis for a Gaussian distribution is 0. *When a function has heavy tails, the kurtosis is positive.*

9.11 Pyramids

Definition 9 *An image pyramid is a collection of representations of an image. The name comes from a visual analogy. Typically, each layer of the pyramid is half the width and half the height of the previous layer. If we were to stack the layers on top of each other, a pyramid would result.*

Definition 10 *A Gaussian pyramid is a pyramid in which each layer is smoothed by a symmetric Gaussian kernel and resampled to get the next layer.*

Definition 11 *A Laplacian pyramid is obtained by taking differences between adjacent layers of Gaussian pyramid. Because the layers of the pyramid don't have the same size, interpolation can be used to increase the size of the coarser image to fit that of the original image. If we hadn't done any subsampling to produce the coarser layer, then one could just take the difference. Note the difference image is equivalent to convolving the original image with a filter which is the difference of two Gaussians (of different variances).*



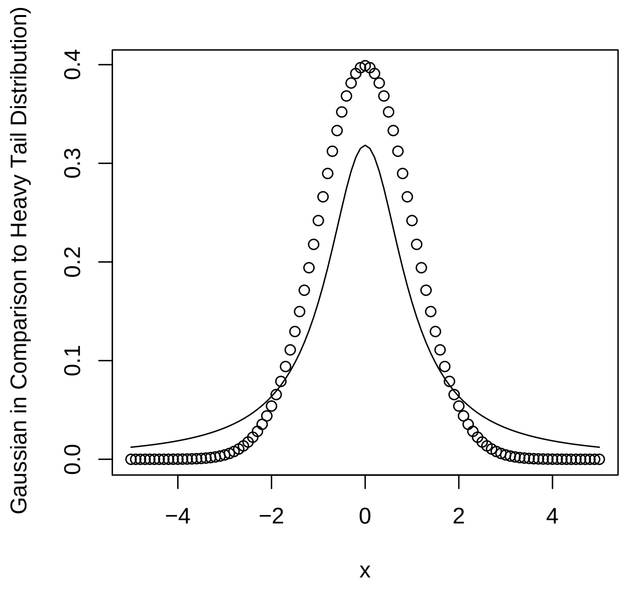


Figure 9.9: Left: Distribution of image regions by area. Right: Heavy Tail Distribution rendered using a solid line; Gaussian Distribution rendered using points

9.12 Optical Flow

We have seen that the movement of the camera with respect to an object gives rise to the optical flow. Today we address the question of the computation of this optical flow. Our input is a video stream $I(x, y, t)$ giving the intensity of a point (x, y) at a time (or frame) t. The output should be the optical flow $u(x, y, t)$, $v(x, y, t)$.

9.12.1 The optical flow constraint equation

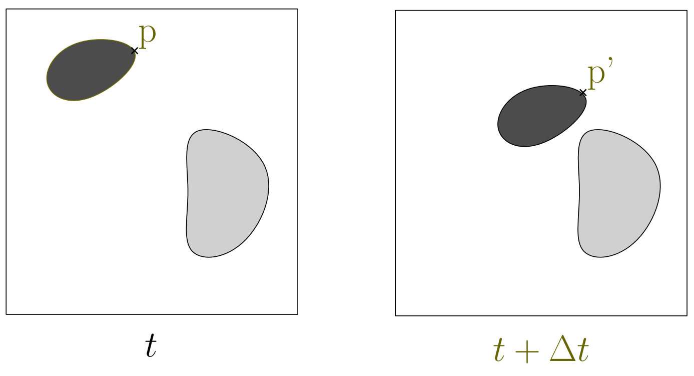


Figure 9.10: The correspondence problem

In order to compute the optical flow, we have to solve the correspondence problem, which is illustrated in Figure 9.10: given a point p in the frame at time t, what is the corresponding point p' in the frame at time $t + \Delta t$? If we can solve this correspondence, then if p has coordinates (x_1, y_1, t_1) and p' has coordinates (x_2, y_2, t_2), we get $u = \frac{x_2 - x_1}{t_2 - t_1}$ and $v = \frac{y_2 - y_1}{t_2 - t_1}$. However in general the correspondance problem is hard to solve. To solve it we make the following key assumption :

Brightness constancy assumption: we assume that the brightness of a given point remains constant over a short period of time.

Clearly this assumption is not always true, as can be seen from the extreme example of a moving light source illuminating a stationary sphere: the brightness on the sphere will change over time, even though it is not moving! By assuming that brightness is constant, we would in that case conclude that the sphere is moving, which is incorrect. However, in most practical cases where the surface is close to lambertian and its movement is slow with respect to the time separating two image frames, it is a very reasonable assumption.

So we assume that $I(x_1, y_1, t_1) = I(x_2, y_2, t_2)$ for corresponding points. By differentiating we immediately get

$$\begin{aligned} \mathrm{d}I &= I_x \mathrm{d}x + I_y \mathrm{d}y + I_t \mathrm{d}t \\ &= 0 \end{aligned}$$

Dividing by dt, this equation can be rewritten as

$$I_x \frac{\mathrm{d}x}{\mathrm{d}t} + I_y \frac{\mathrm{d}y}{\mathrm{d}t} + I_t = 0 \tag{9.8}$$

$$\Longrightarrow I_x u + I_y v + I_t = 0 \tag{9.9}$$

This equation is called the *optical flow constraint equation.* Its unknowns are the optical flow coordinates u and v, whereas I_x, I_y and I_t can be computed from the image. Since we only have one equation in two unknowns, we cannot expect to recover the optical flow from it. This is a frequent problem in computer vision: local equations are not enough to completeley solve the problem. Let us give some intuition on why this is the case here.

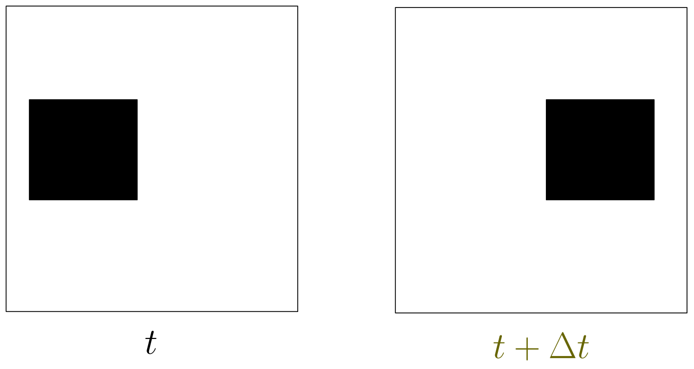


Figure 9.11: A black square moving horizontally

Consider a black square moving horizontally to the right on a white background (see Figure 9.11). Suppose we look at the right edge of this square through a small aperture, then what we see is depicted in Figure 9.12. We have the impression that the edge is moving to the right. However, it could be that the object is actually moving both horizontally and vertically, but we would only be able to detect the horizontal movement through the small aperture: vertical movement would be undetected. This is referred to as the *aperture problem.*

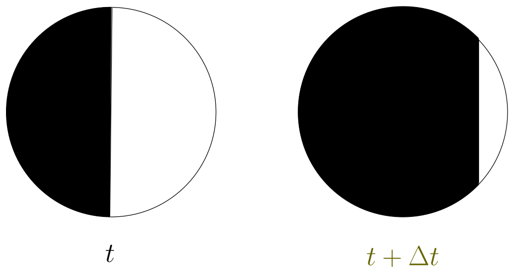


Figure 9.12: The square seen through a small aperture

Equation (9.9) can be interpreted as giving the equation of a line on which the point (u, v) must be, but it could be anywhere on that line. There is one special point (u^*, v^*) on this line, which is the point closest to the origin (see Figure 9.13). This point has the special property of being the point that minimizes the amplitude of the optical flow. It is known that this is the point that is chosen by the human visual system in absence of other cues. For instance, in the example of Figure 9.12, $I_y = 0$, to that the line is vertical of equation $u = -I_t/I_x$. The closest point to the origin on this line is $(-I_t/I_x, 0)$: $v = 0$ so that we do not see any vertical movement.

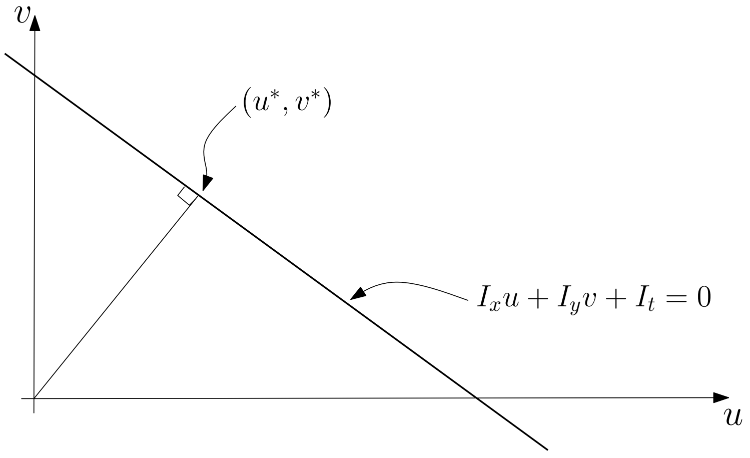


Figure 9.13: The line $I_x u + I_y v + I_t = 0$

9.12.2 The second moment matrix

We rewrite the optical flow constraint equation as

$$\begin{pmatrix} I_x & I_y \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -I_t$$

which can in turn be rewritten in the concise form $\nabla I \cdot u = -I_t$. u now denotes what was previously the vector (u, v).

Suppose we have a neighborhood of n pixels surrounding a point p. We make the following assumption (in addition to the brightness constancy assumption, which we still assume to hold):

The optical flow is constant in a small neighborhood of the point p.

If we define $I^i = I^i(x_i, y_i, t_i)$ to be the intensity of the i-th point in the neighborhood of p, we then have the following equation

$$\begin{pmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \\ \vdots & \vdots \\ I_x^n & I_y^n \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} I_t^1 \\ I_t^2 \\ \vdots \\ I_t^n \end{pmatrix}$$

This equation is of the form $Au = -b$ with

$$A = \begin{pmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \\ \vdots & \vdots \\ I_x^n & I_y^n \end{pmatrix} \qquad \text{and} \qquad b = \begin{pmatrix} I_t^1 \\ I_t^2 \\ \vdots \\ I_t^n \end{pmatrix}$$

We solve it using linear least squares :

$$u = -(A^T A)^{-1} A^T b$$

We have that

$$A^T A = \begin{pmatrix} \sum_i (I_x^i)^2 & \sum_i I_x^i I_y^i \\ \sum_i I_x^i I_y^i & \sum_i (I_y^i)^2 \end{pmatrix} = \sum_i \nabla I^i (\nabla I^i)^T$$

where

$$\nabla I^i (\nabla I^i)^T = \begin{pmatrix} (I_x^i)^2 & I_x^i I_y^i \\ I_x^i I_y^i & (I_y^i)^T \end{pmatrix}$$

is a rank 1 matrix. The matrix $A^T A$ is called the *second moment matrix*. It is an important matrix that shows up in various settings in computer vision. So the general solution to our problem can be expressed in succint form as

$$u = -\left(\sum_i \nabla I^i (\nabla I^i)^T \right)^{-1} \begin{pmatrix} \nabla I^1 & \cdots & \nabla I^n \end{pmatrix} \begin{pmatrix} I_t^1 \\ \vdots \\ I_t^n \end{pmatrix} \tag{9.10}$$

This equation involves the inverse of the second moment matrix. When does this inverse make sense? The second moment matrix is a 2×2 matrix, so it can have rank 0, 1 or 2:

- Rank 0. I is constant over the image : the second moment matrix is then the null matrix. In this case we cannot solve the equations; anyways we wouldn't expect to be able to compute the optical flow in such a case.

- Rank 1. In the example of the black square that we considered before, in the neighborhood defined by our aperture we have that for all points $I_y = 0$, and for some points $I_x \neq 0$ (the points on the border), so that $\sum \nabla I (\nabla I)^T = \begin{pmatrix} K & 0 \\ 0 & 0 \end{pmatrix}$. This matrix is of rank 1 and is not invertible, so we cannot estimate the optical flow. Here, only the normal component (i.e. the component in the direction of the image gradient) can be computed. Indeed, the equation $\nabla I \cdot u = -I_t$ shows that we can estimate the projection of u on ∇I, which is precisely the normal component.

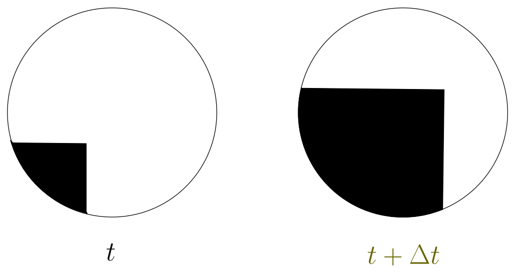


Figure 9.14: A corner seen through a small aperture

- Rank 2. An example where we get a rank 2 matrix is shown in Figure 9.14. The second moment matrices corresponding to the vertical edge of the corner have the form $\begin{pmatrix} K_1 & 0 \\ 0 & 0 \end{pmatrix}$, whereas those corresponding to the horizontal edge have the form $\begin{pmatrix} 0 & 0 \\ 0 & K_2 \end{pmatrix}$, so that when we add these matrices up over all points in the neighborhood, we get a rank 2 matrix that is invertible. This makes sense physically: our intuition is that if we see a corner, we can know precisely it which direction it is moving.

Conclusion: If we have non-zero gradients in different directions, then we can solve the equations (9.10) and recover the optical flow. This condition is morally equivalent to seeing a corner. This is called the Lucas-Kanade technique for estimating the optical flow.

9.13 Orientation/Edge Detection

Consider the image in Figure 9.15. For every pixel in that image, we want to know the direction of the edge at that pixel. Recall that $\nabla I = \begin{pmatrix} I_x \\ I_y \end{pmatrix}$. At a horizontal edge, $\nabla I = \begin{pmatrix} k_1 \\ 0 \end{pmatrix}$, and at a vertical edge, $\nabla I = \begin{pmatrix} 0 \\ k_2 \end{pmatrix}$. In general,

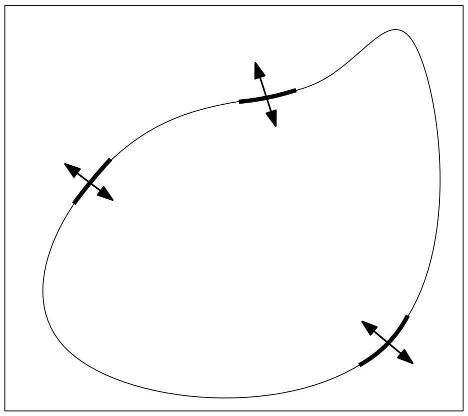


Figure 9.15: Edge orientation detection

we have

$$\frac{\nabla I}{\|\nabla I\|} = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$$

This gives us a $\theta = \theta(x, y)$ at every pixel, which defines the edge orientation at that pixel.

If ∇I is null or very close to 0, then the information given by θ is not reliable: we typically use $\|\nabla I\|$ as a confidence measure, and for $\|\nabla I\|$ below some threshold, we do not declare a direction θ.

Remark 1 *In general we don't exactly compute ∇I, but rather $\nabla(I * G_\sigma)$, i.e. we do some smoothing to eliminate noise.*

9.13.1 The Orientation Histogram

The orientation histogram is a histogram that is defined over some neighborhood (or window) of an image. In this histogram, we plot the number of pixels that have orientation θ for different values of θ : see Figure 9.16 for an example where θ is discretized. This histogram gives a signature of the shape that we are looking at.

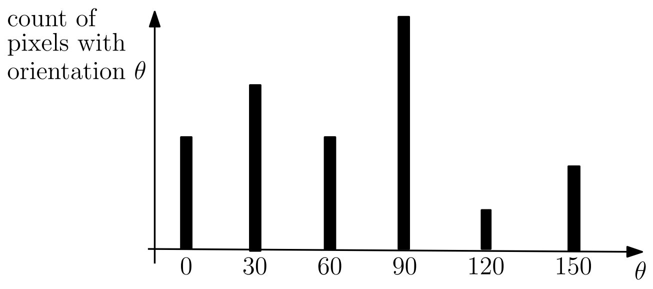


Figure 9.16: A sample orientation histogram

9.13.2 Detecting edges in an image

We describe a technique that is referred to as *non-maximum suppression.* Suppose that we look at a cross-section that is orthogonal to an edge (we know how to do that because of θ). The typical amplitudes of the pixels along the cross-section over the edge are plotted in Figure 9.17. We have a certain threshold on the image gradient over which a given pixel is declared to be an edge pixel. So what would happen in general is that multiple pixels would be marked as edge pixels, but we would like to have a sharp edge, only one pixel wide. For this we use *thinning*, i.e. we look for a pixel that is larger than both its neighbours. This technique is used in the "canny edge detector". Another possible method is to fit a parabola to the pixels along the cross-section, and compute derivatives to find the exact maximum.

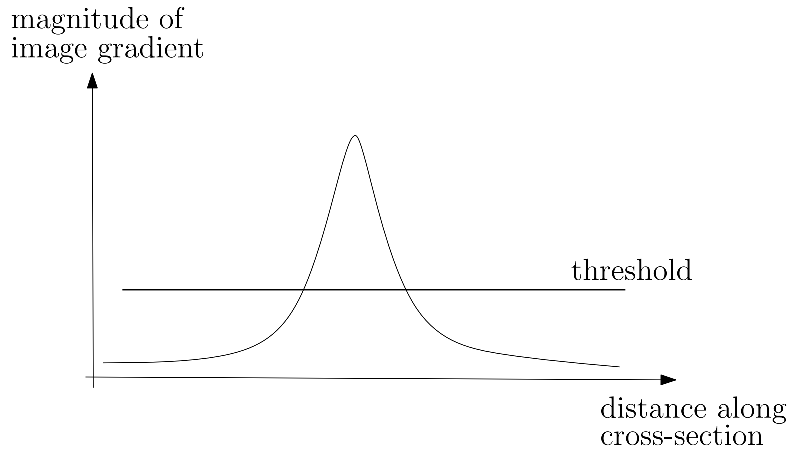


Figure 9.17: Pixel amplitudes along a cross-section

In case we have something that looks like a thick line (zebra stripes for example), then if the line is sufficiently thick the gradient has two distinct spikes, one positive and one negative, and we recognize two edges that border the line. But if the line is relatively thin we get bad results. In that case we would be better off using the second derivative of a gaussian.

By watching a demo we can see that the optical flow is a very powerful cue for breaking camouflage and separating objects from their background (consider the example of a lion chasing a herd of zebras: if all zebras move at the same speed, the lion has a hard time separating one of them from the rest). More on this and texture in the next lecture.