Chapter #

Stereo Vision

Epipolar Geometry, Correspondence and Reconstruction Problems

Zhigang Zhu

Department of Computer Science, The City College of New York, New York, NY 10031

1. INTRODUCTION

The fundamental problem in stereo vision is to infer 3D structure of a scene from two or more images taken from different viewpoints. A pair of stereo images is usually captured by a pair of cameras, thus constructing a stereo pair.

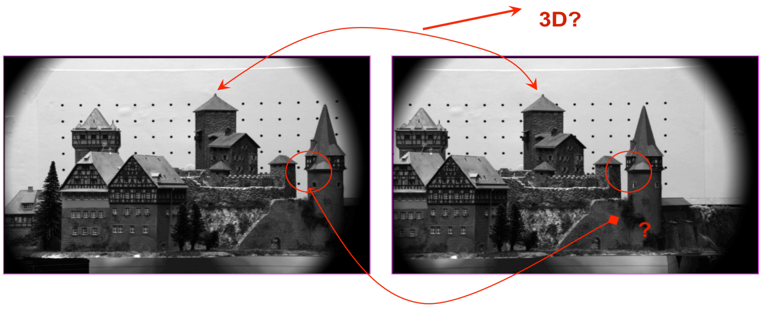


Figure 1. Stereo vision: correspondence and reconstruction problems.

CMU CIL Stereo Dataset : Castle sequence

http://www-2.cs.cmu.edu/afs/cs/project/cil/ftp/html/cil-ster.html

There are two primary sub-problems: the correspondence problem and the reconstruction problem (Figure 1). The correspondence problem (or stereo matching problem) is to find the corresponding points (or correspondences) in a pair of images of each 3D point in the field of view of the stereo cameras and then generate a disparity map. There are two important issues in finding correspondences. First, since the corresponding points of a 3D point in two images are not identical due to the changes in views and illumination, we shall measure if they are "similar" instead of the "same". Second, we will often encounter the occlusion problem: some parts of the scene are visible only through one of the cameras (i.e., eyes).

The reconstruction problem is to find the 3D coordinates of a point given its stereo correspondences. In order to do this, we will need to know some things about theparameters of the two cameras. The reconstruction problem often includes a stereo calibration problem, i.e., to find the camera parameters between the two cameras.

The topics that are going to be covered in this chapter include:

- Stereo vision basics – a simple stereo vision system
- Stereo geometry – epipolar geometry
- Correspondence problem – two classes of approaches
- 3D reconstruction problem – three approaches

2. A SIMPLE STEREO VISION SYSTEM

In this section, we will start with a biomimetic stereo vision system, i.e., the fixated stereo system. Then we will discuss a simple stereo vision system with parallel optical axes. Using this simple stereo vision system, we will derive an important disparity equation, and then analyze the depth resolution of the stereo system. Then we will come back to the fixated stereo system to discuss similar issues as well as its unique properties, such as zero-disparity horopter.

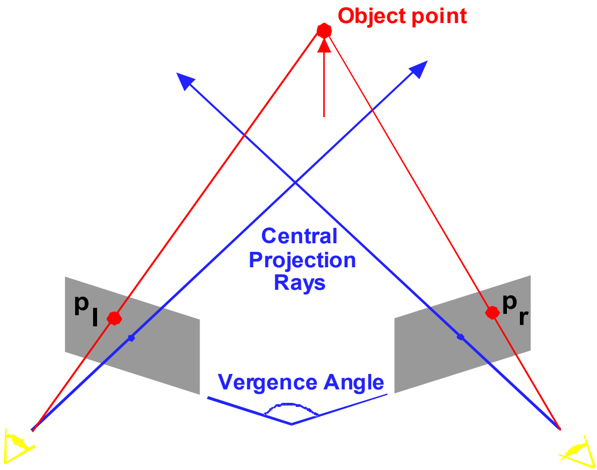


Figure 2. Binocular (or stereo) vision system

The fixated binocular (or stereo) geometry is shown in Figure 2. The vergence angle is the angle between the two image planes, which for simplicity we assume are aligned in such a way that the y-axes are parallel. Given either image of the pair, all we can say is that the object point imaged at p_r or p_l is along the respective rays through the lens center. If in addition we know that p_r and p_l are the image projections of the same object point, then the depth of this point can be computed using triangulation (we will return to this with more details in the section on the simple stereo system with parallel axes).

In addition to the camera geometry, a key additional piece of information in stereo vision is the knowledge that p_r and p_l are projections of the same object point. The correspondence problem in stereo vision is to determine the projection of a point (p_r)in an image given a point (p_l)in the other image. Many solutions to the correspondence problem have been proposed in literature, but none of them have been proven to be entirely satisfactory for general vision (although excellent results can be achieved in many cases).

2.1 Disparity Equation

Now we will derive the disparity equation when the vergence angle is 180 degrees, or the two optical axes are parallel to each other. Figure 3 shows the geometry, where only the XZ plane is drawn. Let us assume that the two cameras only have an offset in the X direction, where the baseline length, the distance between the two cameras, is B. One of the camera coordinate systems, say the left camera coordinate system, will be used as the reference coordinate system to measure the 3D coordinates (X,Y, Z) of a point in space.We also assume that the focal lengths of both cameras are f. Then for a corresponding point pair p_r (x_r,y_r,f) and $p_l(x_l,y_l, f)$, both measuring in their own camera coordinate systems, we have $y_l = y_r$. If we define the disparity in the x direction as

$$d = x_r - x_l \tag{1}$$

we can easily show that the depth (i.e. the Z coordinate of the point P(X,Y,Z)) is

$$Z = D = f\frac{B}{d} \tag{2}$$

This is the important disparity equation. Note that this equation always holds no matter where the point P is located in front of the cameras.

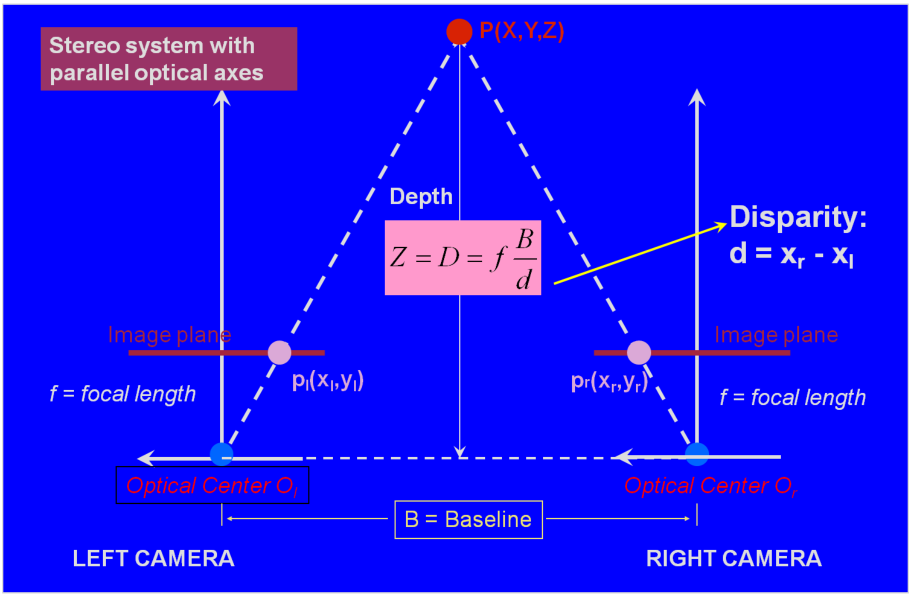


Figure 3. Stereo vision geometry (redraw the figure so P will not be right in the middle)

2.2 Depth Accuracy

It is also important to understand how accurate the depth has been calculated, or the depth resolution, given the same image localization error (i.e. the correspondence error). In Figure 4, the image localization error includes the angles of projection cones that define the localization accuracy of the corresponding points in both the left and right images, i.e., there are localization errors in both x_l and x_r. For simplicity, we can use an overall error in the disparity as the correspondence error, ∂d. Assuming that both the focal length f and the baseline length B are known and without errors, then the depth error can be derived by finding the partial derivative of Z with respect to the disparity d. The absolute depth error can be calculated as

$$\partial Z = \frac{Z^2}{fB}\partial d \tag{3}$$

Note that in equation (3) we only keep the magnitude of the error in Z by removing the negative sign. Readers are encouraged to derive the above equation, noting it will take a few steps to arrive the form of Eq. (3): one of the important steps is to replace the variable *d* in the equation so that it will not show up in the final equation. The relative error can be written as

$$\frac{\partial Z}{Z} = \frac{Z}{fB}\partial d \tag{4}$$

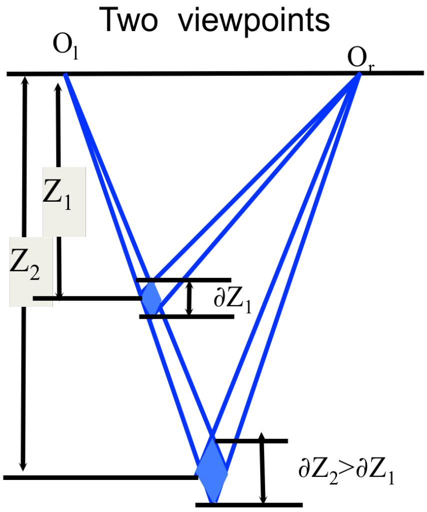


Figure 4. Depth accuracy

From equation (3) we could have the following three observations.

(1) Depth Accuracy (Depth Resolution) vs. Baseline: Depth error is inversely proportional to the baseline length. One of the advantages of a longer baseline is that we could obtain better depth estimation. However, the disadvantages of a longer baseline are that we have a smaller common FOV between the two cameras, and the correspondence problem is harder due to occlusions.

(2). Depth Accuracy (Depth Resolution) vs. Focal Length: Depth error is inversely proportional to the focal length. Longer focal length will provide better depth resolution, but at the same time the cameras will have smaller FOVs.

(3). Depth Accuracy (Depth Resolution) vs. Depth: Depth error is proportional to the square of the depth, indicating that the depth error is a quadrate function of the depth itself. This means that the nearer the point is, the more accurate the depth estimation.

The observation in (3) is also illustrated in Figure 4. We can also use an example to show the decrease of depth accuracy when the distance (depth) of a 3D point increases. Let's assume that the focal length of a camera is f = 16 x 512/8 pixels, i.e. 16 mm focal length with an 8mm wide sensor target and a 512 wide image size. The baseline length is B = 0.5 m. Then the following table shows the relation between depth and depth accuracy, assuming the pixel localization error is 1 pixel. With the normal focal length (16 mm), a decent image resolution (512 pixels), and a large baseline length (0.5 meters), it can be seen that the depth error is very large at a distance of 32 meters, and it is too large to be useful at 128 meters. With a comparable baseline length to human eyes (0.1 meters), the depth errors will be 5 times the values in the table. Therefore, such a stereo vision is basically useless when the depth Z is more than 32 meters.

Table 1. Depth error versus depth

(f = 16 x 512/8 pixels, B = 0.5 m, $\partial(dx) = 1$ pixel, then $\partial Z = Z^2 / 2^9$from Eq. 3)

Z (m)	2	4	8	16	32	64	128	256
∂Z(m)	1/128	1/32	1/8	½	2	8	32	128

2.3 Stereo with Converging Cameras

We have seen that the stereo vision system with parallel optical axes has a dilemma (Figure 5): a shorter baseline length can provide a larger common FOV between the two cameras, but will produce a larger depth error. On the other hand, a longer baseline length will yield a smaller depth error, but unfortunately it will result in a smaller common FOV and increased occlusion problems.

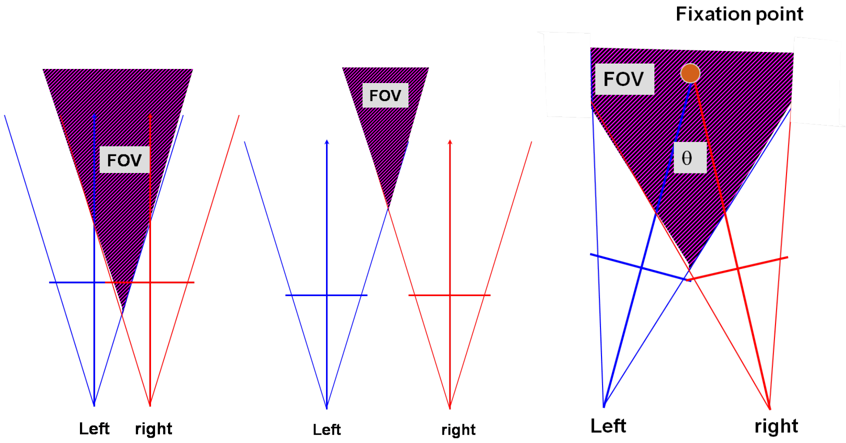


Figure 5. Stereo with converging axes

To break this dilemma, one solution is to use a stereo vision system with converging cameras (Figure 5 right), where two optical axes intersect at a fixation point in space, where the two image planes form a vergence angle shown in Figure 1. In this setting, the common FOV increases with the same large baseline as in the setup in the middle of the figure. For the convenience of discussion, we define converging angle as θ, as shown in Figure 5.

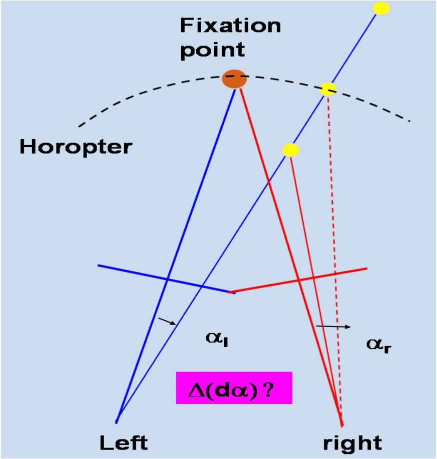


Figure 6. Stereo horopter

We could also derive some interesting disparity properties for the stereo vision system with converging axes as we did for the stereovision system with parallel optical axes. Here the disparity is defined in angles instead of distances, as

$$d\alpha = \alpha_r - \alpha_l \tag{5}$$

(1). A zero disparity is created at the fixation point, and in fact a zero-disparity surface in space, which is called *zero-disparity horopter,* can be defined when $\alpha_r = \alpha_l$.

(2) The magnitudes of disparity values increase with the distances of object points from the fixation point (and the zero-disparity horopter). Clearly we have dα > 0, when a point is outside of the horopter and dα < 0 when it is inside the horopter.

(3). Depth Accuracy vs. Depth: The fixation stereo does not change the fact that the depth resolution is still proportional to the square of depth (Question 2).

3. EPIPOLAR GEOMETRY

In this section, we will formally discuss the epipolar geometry of a general stereo vision system. We will first define epipolar lines – to determine where to search correspondences, from the formation of the epipolar plane and epipoles. Then we will introduce two important matrices in stereo vision: the essential matrix E and the fundamental matrix F. We will give an Eight-Point Algorithm to find the fundamental matrix, and discuss how to compute the epipoles. Finally we will discuss stereo rectification to ease the correspondence problem.

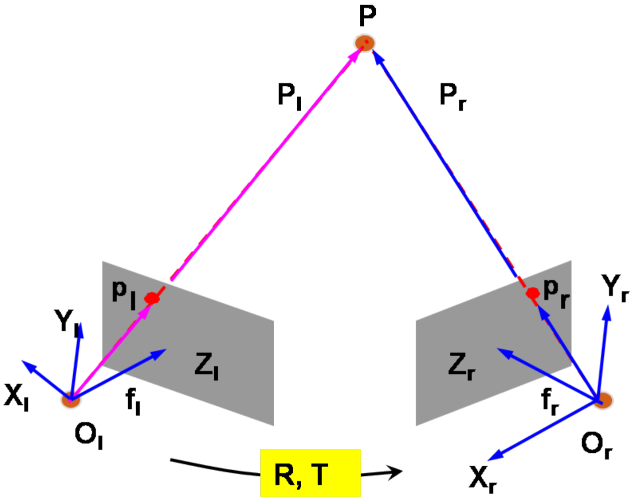


Figure 7. Parameters of stereo system

3.1 Parameters of a Stereo Vision System

A stereo vision system can also be specified by a set of intrinsic parameters and a set of extrinsic parameters. The intrinsic parameters characterize the transformation from camera to pixel coordinate systems of each camera, including the focal lengths, image centers, aspect ratios of the two cameras. In the following, we will only use the two focal lengths, f_l and f_r, as parameters, assuming the image coordinates are all measured in their image coordinate systems. The extrinsic parameters describe the relative position and orientation of the two cameras, and can be represented by the rotation matrix R and translation vector T.

For a 3D point P, its representations in the left and right camera coordinate systems are noted as $P_l = (X_l, Y_l, Z_l)$ and $P_r = (X_r, Y_r, Z_r)$. They are the vectors of the same 3-D point P, represented in the left and right camera coordinate systems respectively. We have

$$\mathbf{P_r = R(P_l - T)} \qquad (6)$$

The extrinsic parameters have obvious meanings. The translation vector T is simply a vector from the centers of the left camera to the right camera, $(O_r - O_l)$, measured in the left camera coordinate system. The rotation matrix R represents the relative rotational relation between the two cameras.

Define $p_l = (x_l, y_l, z_l)$ and $p_r = (x_r, y_r, z_r)$ as projections of P on the left and right image planes respectively. For all image points, if we note $z_l = f_l$, $z_r = f_r$, then we can represent the perspective projection equations of the two cameras in matrix forms:

$$\mathbf{p}_r = \frac{f_r}{Z_r}\mathbf{P}_r \, , \, \mathbf{p}_l = \frac{f_l}{Z_l}\mathbf{P}_\mathbf{l} \qquad (7)$$

Usually we define the left camera as the *reference* camera. The translational vector T is represented in the left camera system, the rotation matrix is defined as a transformation from the left to the right cameras.

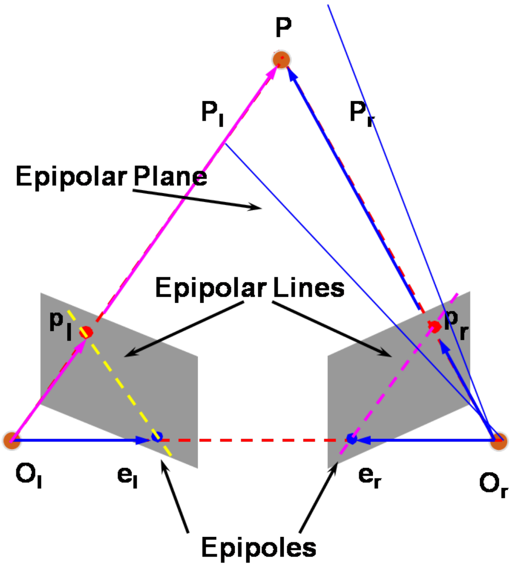


Figure 8. Epipolar geometry

The main purpose of establishing the epipolar geometry is to define where to search for correspondences. For a 3D point P, its epipolar plane is defined as a plane going through the point P and the centers of projections (COPs) of the two cameras, O_l and O_r. The two conjugated epipolar lines are the lines where the epipolar plane intersects the two image planes, whereas each of the two epipoles (e_l and e_r) is the image projection of the COP of one camera onto the other.

The epipolar constraint can be stated as: corresponding points must lie on conjugated epipolar lines.

Note that given two cameras, O_l and O_r, for any image point p_l in the reference (left) image, an epipolar plane can be defined by these three points: O_l , O_r and p_l. Then, an epipolar line can be defined as the intersection of the epipolar plane with the right image plane, as the dashed pink line in Figure 8, and the corresponding point p_r must lie on the epipolar line. In other words, if the stereo system is calibrated, i.e., we know all the intrinsic and extrinsic parameters, the equation of the epipolar line in the right image of any point in the left image can be written out, by using plane intersection.

However, we are more interested in understanding how many parameters we really need to know in order to define an epipolar line. So the goal is to build up a relation between the two corresponding points and then analyze what we need to know to define their conjugated epipolar lines.

3.2 Essential Matrix

Note that the vectors **T** = (**O**$_r$-**O**$_l$), ***P$_l$*** and ***(P$_l$-T)*** are on the same plane (the epipolar plane of the point ***P***), all represented in the left camera coordinate system. The projection of the vector ***T***×***P$_l$*** on the vector (P$_l$-T) is zero, since they are orthogonal to each other. So we have:

$$(\mathbf{P}_l - \mathbf{T})^T \mathbf{T} \times \mathbf{P}_l = 0 \tag{8}$$

From equation (6) we have

$$(\mathbf{P}_l - \mathbf{T})^T = \mathbf{P}_r^T \mathbf{R}.$$

Plugging this into equation (8) and defining an Essential Matrix

$$E = RS \tag{9}$$

where,

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

We will have,

$$\mathbf{P}_r^{\mathbf{T}} \mathbf{E} \mathbf{P}_l = 0$$

Note that the essential matrix is 3x3 matrix constructed from R and T. Since rank(S) = 2, we have rank (E) = 2, and E has two equal nonzero singular values. By using the relations in equations in (7), we have:

$$\mathbf{p}_r^{\mathbf{T}} \mathbf{E} \mathbf{p}_l = 0 \tag{10}$$

This gives a natural link between the stereo point pair in the two images, and the extrinsic parameters of the stereo system. We can make the following two observations from equation (10):

(1) A pair of correspondence (**p$_l$**, **p$_r$**) provides a linear equation of 9 entries of the essential matrix E, therefore given 8 pairs of correspondence points, we can obtain an estimation of the essential

matrix. We will provide a detailed algorithm later in the section for estimating the fundamental matrix, which can be used to estimate the essential matrix as well.

(2) Equation (10) provides the mapping between a pair of correspondence points and their epipolar lines that we are looking for. For example, given p_l and E, equation (10) represents a line equation of p_r in the right plane.

Note that p_l, p_r are in their corresponding camera coordinate systems, not pixel coordinates that we can directly measure. Therefore we have to know the intrinsic parameters of the two cameras in order to use equation (10) for deriving the epipolar line relation. From observation (1), it means that we find the epipolar geometry relation of the stereo system given eight pairs of images points if the intrinsic parameters are known.

Here we provide more analysis of epipolar line geometry.

The essential matrix equation represents the epipolar plane in either the left or the right image. On one hand it represents the epipolar line in the right image, of a point in the left image, p_l

$$(Ep_l)^T p_r=0$$

where $p_r = (x_r,y_r,f_r)^T$ is a column vector representing the corresponding point in the right image, and $(Ep_l)^T$ is a row vector of three coefficients. On the other hand, it also represents the epipolar line in the left image, of a point on the right image, p_r

$$(p_r^T E) p_l=0$$

where pl = $(x_l,y_l,f_l)^T$ is a column vector representing the corresponding point in the left image, and $(p_r^T E)$ is a row vector of three coefficients.

We also note that the epipolar line about p_r in the right image will pass through the location p_l if and only if the rotation matrix is an identity matrix, i.e., R = I , which says

$p_l^T S\, p_l == 0$ [Question 3. check if this is correct]

We will see in the next chapter that this is actually the focus of expansion (FOE) geometry in visual motion. If the rotation matrix R is not an identity matrix, then the epipolar line for p_r will not pass through the point p_l, if putting both of them in the right image.

3.3 Fundamental Matrix

Now we are ready to derive the mapping between a pair of correspondence points and their epipolar lines in the two pixel coordinate systems, without knowing the intrinsic parameters.

We will need to derive a matrix representation M of intrinsic parameters of a camera, including f, s_x, s_y, o_x and o_y. We know that $f_x = f/s_x$, $f_y = f/s_y$, then for an image point $p = (x,y,f)^T$ in the camera coordinate system, we have its pixel representation as

$$\begin{aligned}\bar{\mathbf{p}} &= (x_1, x_2, x_3)^T \\ &= (-f_x x + f o_x, -f_y y + f o_y, f)^T = (-x/s_x + o_x, -y/s_y + o_y, 1)^T\end{aligned}$$

Or in the matrix form as

$$\bar{\mathbf{p}} = \mathbf{M}\ \mathbf{p}$$

where

$$\mathbf{M}_{\text{int}} = \begin{bmatrix} -f_x & 0 & o_x \\ 0 & -f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (11)$$

and it's corresponding pixel coordinates are $x_{im} = x_1/x_3$, $y_{im} = x_2/x_3$.

Now let us define M_l and M_r as the intrinsic matrices of the left and the right cameras, respectively, then we have:

$$\mathbf{p}_r = \mathbf{M}_r^{-1}\bar{\mathbf{p}}_r \quad \mathbf{p}_l = \mathbf{M}_l^{-1}\bar{\mathbf{p}}_l \qquad (12)$$

Inserting equation (12) into equation (10) we have the following fundamental matrix equation

$$\bar{\mathbf{p}}_\mathbf{r}^{\mathbf{T}}\mathbf{F}\bar{\mathbf{p}}_\mathbf{l} = 0 \qquad (13)$$

where the fundamental matrix can be written as,

$$\mathbf{F} = \mathbf{M}_r^{-\mathbf{T}}\mathbf{E}\mathbf{M}_l^{-1} \qquad (14)$$

where –T represents both the transpose and the inverse (of the matrix M_r). This includes both intrinsic and extrinsic parameters in a 3x3 matrix. Looking into equation (13) we see that

$$\begin{pmatrix} x_{im}^{(r)} & y_{im}^{(r)} & 1 \end{pmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{pmatrix} x_{im}^{(l)} \\ y_{im}^{(l)} \\ 1 \end{pmatrix} = 0 \qquad (15)$$

We can summarize the following observations about the Fundamental Matrix.

(1) Rank (F) = 2.
(2) It encodes information on both intrinsic and extrinsic parameters.
(3) It enables full reconstruction of the epipolar geometry.
(4) It is in the pixel coordinate systems without the need of knowing any knowledge of the intrinsic and extrinsic parameters.
(5) A pair of corresponding points provides a linear equation of the 9 entries of F.

Below we provide the algorithms to calculate the fundamental matrix F, and epipoles (Figure 8). Similarly we can compute **E** given intrinsic parameters.

[Normalizing: The Points]

The reason for the normalization is to balance the coefficients in the linear equation system so that it could be less ill-conditioned.

- Input: n points (n >= 8)
    - Find the mean value of the x and y coordinates of the entire group of points.
    - Find the scaling factor by dividing the sqrt(2) by the hypotenuse of the x and y means.
    - Create a 3x3 translation matrix in the form of: $$T = scale \times \begin{bmatrix} 1 & 0 & -x_{mean} \\ 0 & 1 & -y_{mean} \\ 0 & 0 & \frac{1}{scale} \end{bmatrix}$$
    - Translate all of the points by doing $T \times p$
- Output: n Normalized points

[Computing F: The Eight-point Algorithm]

- Input: n point correspondences (n >= 8)
    - Construct homogeneous system **Ax** = 0 from $\bar{\mathbf{p}}_{\mathbf{r}}^{\mathbf{T}} \mathbf{F} \bar{\mathbf{p}}_{\mathbf{l}} = 0$
        - **x** = (f_{11},f_{12}, f_{13}, f_{21},f_{22},f_{23} f_{31},f_{32}, f_{33})^T : entries in F

    - Each correspondence give one equation
    - **A** is a nx9 matrix
  - Obtain estimate $\hat{\mathbf{F}}$ by SVD of A $\mathbf{A} = \mathbf{UDV}^T$
    - **x** (up to a scale) is the column of **V** corresponding to the least singular value
  - Enforce singularity constraint: since Rank (**F**) = 2
    - $\hat{\mathbf{F}} = \mathbf{UDV}^T$ Compute SVD of $\hat{\mathbf{F}}$
    - Set the smallest singular value to 0: **D** -> **D'**
    - $\mathbf{F'} = \mathbf{UD'V}^T$ Correct estimate of **F** :
- Output: an estimate of the fundamental matrix, **F'**

[Denormalizing F:]

- Input: Normal Fundamental Matrix $\mathbf{F_N}$, Translation Matrixes $\mathbf{T_l}$, $\mathbf{T_r}$
  - $F = T_r^T F_N T_l$
- Output: Regular Fundamental Matrix **F**

When calculating the fundamental matrix of a pair of images, a better estimate will require more than 8 pairs of points. The chosen points should also be scattered all around the image to get a uniformly accurate fundamental matrix.

[Locating the Epipoles from F]

- Input: Fundamental Matrix **F**
  - Find the SVD of **F**
  - The epipole $\mathbf{e_l}$ is the column of **V** corresponding to the null singular value (as shown above)
  - The epipole $\mathbf{e_r}$ is the column of **U** corresponding to the null singular value
- Output: Epipole $\mathbf{e_l}$ and $\mathbf{e_r}$

A few notes on estimating epipoles (Figure 8):

(1) Epipole on the left image

First, let us define the dot product of two vectors $\mathbf{p_r}$ and $\mathbf{Fe_l}$ as a scalar value q_l. Since **F** is a non-zero matrix, $\mathbf{p_r}$ could be anything, and q_l must be zero (from Equation (13), then we know $\mathbf{Fe_l}$ has to be a zero vector: i.e., $\mathbf{Fe_l} = \mathbf{0}$. Multipling the transpose of F to both sides, we have a homogeneous equation system.

$$\mathbf{F}^\mathbf{T}\,\mathbf{Fe_l} = 0$$

By applying SVD, $\mathbf{F} = \mathbf{UDV}^\mathrm{T}$, we have the columns of V as the eigenvectors of $\mathbf{F}^\mathrm{T}\mathbf{F}$, therefore the solution is the eigenvector corresponding to the null eigenvalue 0.

(2) Epipole on the right image
Do the same thing to $\mathbf{e_r}$.

3.4 Stereo Rectification

The simple stereo system with parallel optical axes has both its epipoles at infinity. Note that for such a stereo vision system, R = I, and T =(T_x,0,0) = (B,0,0). Therefore all the epipolar lines are horizontal scan lines such that y_r=y_l.

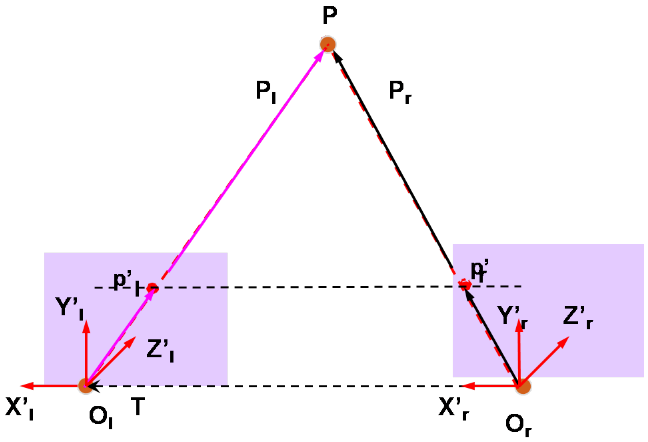


Figure 9. Epipolar lines of stereo vision with parallel optical axes

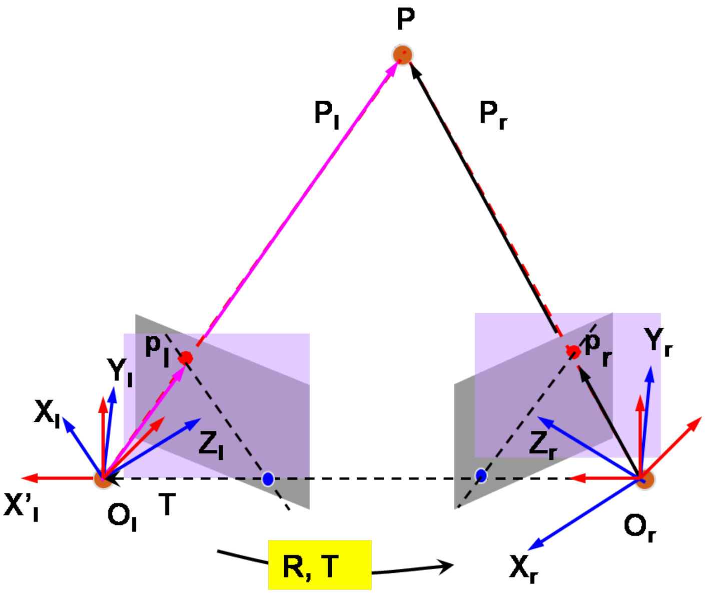
We can arrive at the same conclusion from equation (10). Since R = I, we have

$$\mathbf{P_r}^T\mathbf{S}\mathbf{P_l} = 0 \qquad (16)$$

with the matrix S that have (T_x, T_y, T_z) = (B, 0, 0), where B is the baseline length. That is

$$\begin{bmatrix} x_r & y_r & f_r \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -B \\ 0 & B & 0 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ f_l \end{bmatrix} = 0 \qquad (17)$$

If $f_l=f_r$, we can easily arrive at the same result w $y_r=y_l$. The horizontal epipolar scanline constraint makes the simple stereo vision system very attractive particularly for hardware implementation since the search for the corresponding point of any given point in the reference image is along its horizontal scanline.



$$X_l' = T, \quad Y_l' = X_l' x Z_l, \qquad Z'_l = X_l' x Y_l'$$

Figure 10. Stereo rectification

Since we have this advantage, it is very desirable to perform an image rectification of the stereo pair so that given the stereo pair and its intrinsic and extrinsic parameters; we can find an image transformation to achieve a stereo system with horizontal epipolar lines.

Here we discuss a simple algorithm, assuming the stereo cameras have been calibrated.

[Stereo Rectification Algorithm]

Step 1. Rotate the left camera so that its new X axis (X') points to the direction of T, i.e., the vector point from the right camera center to the left camera center. One solution is to find an orthogonal coordinate system $X_l'Y_l'Z_l'$ by finding three unit vectors pointing towards its three axes, $\mathbf{X_l'} = \mathbf{T}/|\mathbf{T}|$, $\mathbf{Y_l'} = \mathbf{X'} \times \mathbf{Z_l}$ where $\mathbf{Z_l}=(0,0,1)^T$, and $\mathbf{Z_l'}=\mathbf{X_l'}\times \mathbf{Y_l'}$. Then the rotation matrix to rectify the left camera can be defined as $\mathbf{R_{rect}} = (\mathbf{X_l'}, \mathbf{Y_l'}, \mathbf{Z_l'})$:

$$\mathbf{P_l'} = \mathbf{R_{rect}}\, \mathbf{P_l} \qquad (18)$$

Step 2. Rotate the right camera so that its three axes have the same directions as those of the left camera. From equation (6), we can see that the rotation matrix for rectifying the right camera is $\mathbf{R_{rect}R^T}$:

$$\mathbf{P_r' = R_{rect}\, R^T\, P_r} \tag{19}$$

In the rectified stereo vision system, we have

$$\mathbf{P'_r = P'_l - T'} \tag{20}$$

where $\mathbf{T' = R_{rect}R}$, which is in the form of $\mathbf{(B, 0, 0)^T}$,

Step 3. The stereo rectification can be implemented by software image transformations instead of actually rotating the two cameras as shown in equations (18) and (19). As we have discussed in the camera model chapter, a rotational transformation can be implemented as an image rectification. For example, for the left camera, the transformation for the image rectification is

$$\mathbf{p}_l^{'} \cong \mathbf{R}_{rect}\mathbf{p}_l \tag{21}$$

where the equality is a projective equality.

3.5 Epipolar Geometry Summary

We have discussed the epipolar geometry in details. The purpose is to define where to search for correspondences. Here is a brief summary of what we have learned.

(1) ***About epipolar plane, epipolar lines, and epipoles***: depending on the knowledge of the intrinsic and extrinsic parameters, we can do the following:

- known intrinsic (f) and extrinsic (R, T)
    - co-planarity equation (8)
- known intrinsic but unknown extrinsic
    - essential matrix (10)
- unknown intrinsic and extrinsic
    - fundamental matrix (13)

(2) ***About stereo rectification***:

- Generate a stereo pair (by software) with parallel optical axis and thus achieving horizontal epipolar lines, from an arbitrary stereo vision setup.

4. CORRESPONDENCE PROBLEM

We have three questions to ask when we deal with the correspondence problem:

(1) What to match? Should the *features* be points, lines, areas, or structures?

(2) Where to search for correspondence? We know the corresponding points shall be searched along epipolar lines, but what else shall we consider?

(3) How to measure similarity? We know the correspondence points are usually not identical, they are just similar. Depending on features, what kinds of similarity measures could we use?

In this section, we will discuss two basic approaches: the correlation-based approach and the feature-based approach. After discussing these two topics, we will briefly mention some advanced topics that have been proposed to improve the performance of stereo vision algorithms, including:

- Image filtering to handle illumination changes
- Adaptive windows to deal with multiple disparities
- Local warping to account for perspective distortion
- Sub-pixel matching to improve accuracy
- Self-consistency to reduce false matches
- Multi-baseline stereo

4.1 Correlation-based approach

The basic steps for the correlation-based stereo matching approach are defined as: For each point (x_l, y_l) in the left image, define a window centered at the point, then search its corresponding point within a search region in the right image. The disparity (dx, dy) is the displacement when the correlation is maximum. Figure 11 illustrates the idea.

As we have mentioned, there are three important issues in stereo matching.

(1). Elements to be matched. In the correlation-based approach, an image window of fixed size centered at each pixel in the left image is defined as the matching element for the pixel.

(2). Similarity criterion. This is the measure of similarity between windows in the two images. The corresponding element is given by the window that maximizes the similarity criterion within a search region; we will provide more details below.

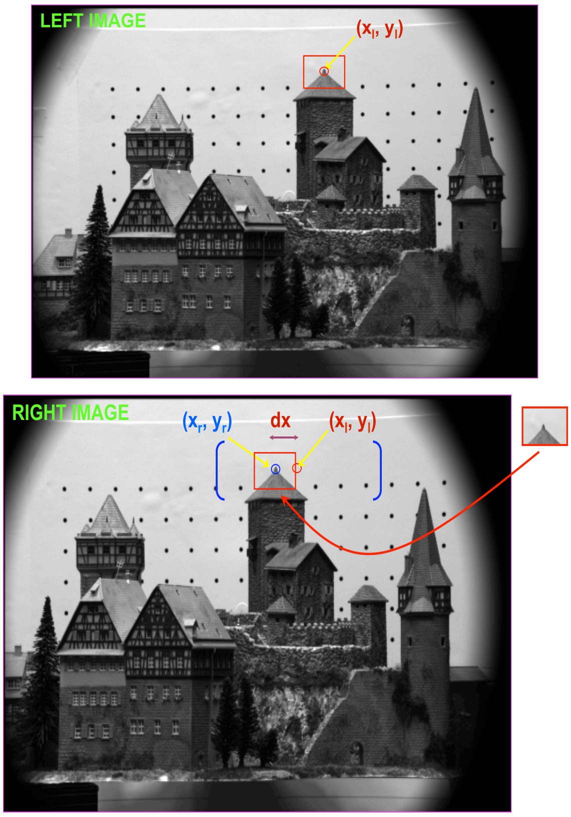


Figure 11. Correlation-based approach

(3). Search regions. Theoretically, search regions can be reduced to a 1-D segment, along the epipolar line, and within the disparity range, given that the depth range is known. In practice, we search a slightly larger region both along and normal to the epipolar line, due to errors in calibration.

With these, we formally write the correlation between two WxW windows, in the left image and the right image, respectively, as

$$c(d_x, d_y) = \sum_{k=-W}^{W} \sum_{l=-W}^{W} \psi(I_l(x_l + k, y_l + l), I_r(x_l + d_x + k, y_l + d_y + l)) \quad (22)$$

where Ψ is a similarity measure function. The similarity criterion typically has the following three forms:

(1) Cross-Correlation

$$\Psi(u, v) = uv \quad (23)$$

(2) Sum of Square Difference (SSD)

$$\Psi(u,v) = -(u-v)^2 \tag{24}$$

(3) Sum of Absolute Difference (SAD)

$$\Psi(u,v) = -|u-v| \tag{25}$$

After calculating all of the similarity values in the search range $\{(d_x,d_y)\}$, the final disparity vector is defined as:

$$\bar{\mathbf{d}} = (\bar{d}_x, \bar{d}_y) = \arg\max_{\mathbf{d}\in R}\{c(d_x,d_y)\} \tag{26}$$

This simple approach has both pros and cons that we list below:

PROS:

(1) It is easy to implement, both in software and hardware.
(2) It produces a dense disparity map, for each pixel.
(3) It might be slow without optimization, but the algorithm can be implemented in parallel, e.g., using GPUs.

CONS:

(1) It needs highly textured images to work well.
(2) It is inadequate for matching image pairs from very different viewpoints, due to illumination changes.
(3) Windows may cover points with quite different disparities, thus producing blurs on depth changes.
(4) It produces inaccurate disparities on the occluding boundaries.

A stereo pair of a campus scene at UMass Amherst (Figure 12) illustrates all the issues of this approach: regions with less or no texture, depth boundaries between two objects, and different occlusions due to view changes. For example, the centers of the blue boxes within the red circles are corresponding points, but the rectangular windows cover two three planar surfaces in the left image and two surfaces in the right image: the side of the building is occluded in the right image. Even without this problem, the two windows still cover multiple depths that make the simple window-based correlation problematic. As a [Project], please check out all the issues you could find in these two images.

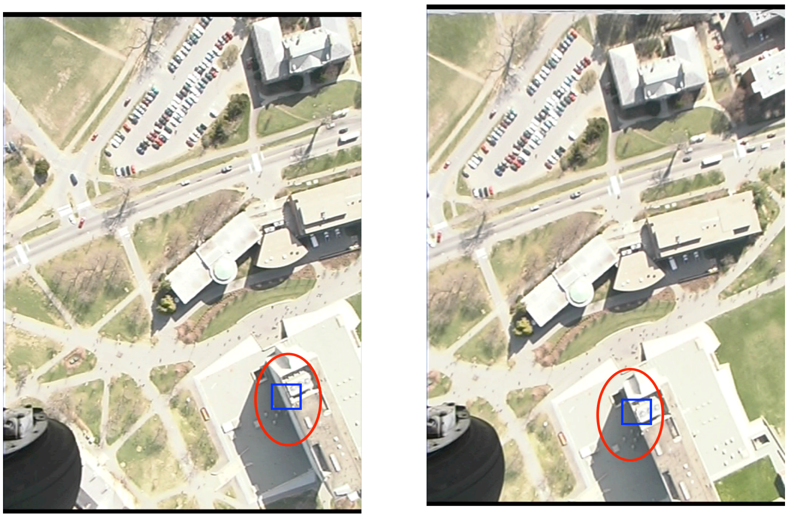


Figure 12. Problems in stereo correspondences

4.2 Feature-based Approach

4.2.1 Basic Ideas

Matching primitives in feature-based approaches can be the following (Figure 13):

- Edge points: Points defined on those points that have high gradient magnitudes and thus, can be declared as edge points.
- Lines: Line segments extracted by linking edge points via either edge tracking or Hough Transform, with their attributes, such as length, orientation, average contrast, etc.
- Corners: Points on contours that have high curvatures, for example Harris corners [REF]
- Structures: Higher-level structures that are formed by group edge points, lines, and/or corners. For example, the closed contour of a surface.

A matching algorithm usually includes the following three steps.

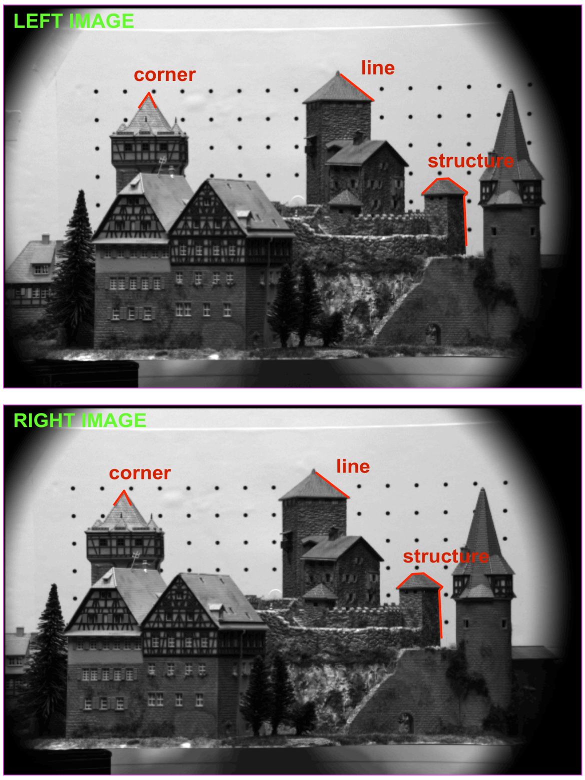


Figure 13. Feature-based matching

Step 1. Extract features in the stereo pair. For example, if the matching primitives are line segments, an edge detector such as Canny edge detector can be used to detect edges. Line segments can be formed by a line tracking and fitting approach or a Hough Transform based approach. A line segment can be defined by its endpoints, length, orientation, and average contrast.

Step2. Define similarity measure. For example, if the features are line segments as defined in Step 1, the similarity measure between two line segments in two images (respectively) could be derived from their differences in orientation, length and contrast.

Step 3. Search correspondences using similarity measure and the epipolar geometry. We expect that only a few candidates of a feature in the reference image are within the search ranges in the right image, and the candidate with the highest similarity values is picked up as the correspondence.

For each feature in the left image, search in the right image; the disparity (d_x, d_y) is the displacement when the similarity measure is at maximum. As in the correlation approach, the feature-based approach also has its pros and cons.

PROS:

(1) It is relatively insensitive to illumination changes.
(2) It is good for man-made scenes with strong lines against surfaces with weak or no texture.
(3) It works well on the occluding boundaries (edges).
(4) It could be faster than the correlation approach.

CONS:

(1) It only creates a sparse depth map, therefore many points in the reference image may not have depth values.
(2) Feature extraction may be tricky. For example, lines (edges) might be partially extracted in one image. It is also hard to know what the best way is to measure the similarity between two lines, since perspective distortion and feature extraction features in two images may be very different.

4.2.2 Feature detection

[Moved Section 4.1.1 from Visual Motion Chapter to here]

In computer vision, a local *feature* represents a group of local information in either 2D or 3D space that is distinctive and easy to identify. It usually contains some special local image properties, such as a corner, a line, a textured region or a planar surface in a range image. Some are scale and orientation invariant, such as Harris corners and SIFT features so that they can be matched between two images with a large perspective distortion. In this section, some popular image features are briefly introduced. These features can be used in matching both stereo and motion images.

Line features

Many researchers use line segments as image features. In 2D image space, line segments can be extracted from the edge map of an image. They are reliable since edges are relatively invariant to projection distortion, even under illumination changes. The problem is that it's hard to distinguish the true endpoints of an edge segment. A line feature descriptor may include the following values (Trucco and Verri, 1998): the length of the line, l, the orientation of line, θ, the midpoint of the line, $m=[x, y]^T$ and average contrast along the edge line c.

Similarity match between two line feature descriptors is as following:

$$M = \frac{1}{w_0(l_l - l_r)^2 + w_1(\theta_l - \theta_r)^2 + w_2 \mid m_l - m_r \mid_2^2 + w_3(c_l - c_r)^2} \quad (27)$$

where $w_{0,}$ $w_{1,}$ w_2 and w_3 are weights and the subscripts *l* and *r* refer to the left and right images, respectively.

Corner features

A corner is a 2D image feature point that is the intersection of two edges. It is a popular "interest point" detector due to its strong invariance to rotation, scale, illumination and insensitivity to image noises. Consider the spatial image gradients, including gradients in both the horizontal and vertical directions. A covariance matrix C that represents the likelihood of a corner feature, is defined as

$$C = \begin{bmatrix} \sum E_x^{\ 2} & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^{\ 2} \end{bmatrix} \quad (28)$$

where E_x and E_y are the gradients in the vertical and horizontal directions, respectively. C is actually the least square form of the coefficient matrix of the spatial gradient evaluation (the 2x2 matrix A^TA).

The following local image properties around the pixel *(x,y)* can be obtained by the eigen value analysis of matrix C (Figure 14).

(1) If both eigen values of C are small, then the pixel *(x,y)* is in an area with a rather uniform texture;

(2) If one of the eigen values is small and the other is a large positive number, then the pixel *(x,y)* is close to an edge;

(3) If both eigen values are large positive numbers, then the pixel *(x,y)* is around the intersection of edges, i.e., at a corner.

Therefore, the corner detection can be achieved by enforcing a minimal value on the smallest eigen value (Shi and Tomasi 1994). It can also be achieved using the following corner response function $\det(C) - k \times trace(C)^2$ (Harris and Stephens 1986) where it does not need the complicated eigen-analysis processing, and *k* is a constant (0.04-0.15) obtained from experiments. This method is the well-known "Harris corner" method. The corner response is used to determine whether a local region is a corner feature by comparing it with a threshold. The Harris corner detector can be of different sizes, from 3x3 to 15x15 and a match is usually performed with a correlation window.

Multi-scale as well as scale and affine invariant extensions of corner feature, Harris-Laplace and Harris-Affine are also proposed (e.g., in Mikolajczyk and Schmid 2004).

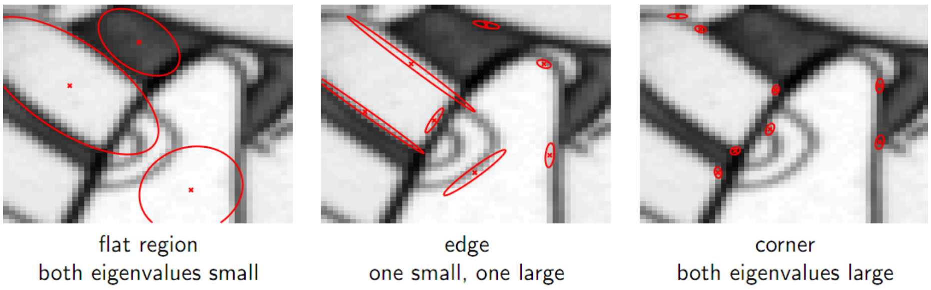


Figure 14. Relation between local feature properties and eigen analysis of matrix C

SIFT features

Though Harris corner and its variations have been widely used in different image tracking and matching tasks, they cannot handle large perspective distortion. Lowe (Lowe 2004) proposed scale invariant feature transformation (SIFT) to address this issue. It takes an image and generates a large collection of feature vectors, which are invariant to the scaling, rotation or translation of the image. The SIFT feature extraction procedure includes the following four steps: (1) a scale space extrema detection; (2) key point localization; (3) orientation assignment; and (4) SIFT descriptor construction.

a. *Scale-space extrema detection*

In the first step, a 'scale-space' function (variant scales in the frequency domain) is used to identify the locations and scales that are invariant to different perspective views of the same "object". The scale-space function is defined as

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{29}$$

where the operator * is a convolution operation, $G(x, y, \sigma)$ is a variable-scale Gaussian with a scale variable σ, and I(x, y) is the input image. To locate scale-space extrema, a difference of Gaussians is applied,

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{30}$$

which is the difference between two nearby scales separated by the factor k. To detect the local maxima and minima of $D(x, y, \sigma)$, each point is compared with its 8 neighbors at the same scale, and its 9 neighbors up and down one scale. If this value is the minimum or maximum of all these points then this point is an extremum.

b. Key point localization

In order to eliminate the key points obtained by the first step in regions with low contrasts or around edges, the second step first computes the value of the second-order Taylor expansion of Difference-of-Gaussian (DOG) $D(x, y, \sigma)$ at any key point; the key point with small value is filtered out, and then the principal curvatures of the DOG image in two directions, one along the direction of edge and the other in the perpendicular direction, is evaluated at each key point. If the two curvatures have a large difference (i.e., the ratio between the larger and the smaller is high and it represents an edge) and the key point is close to an edge, then it’s eliminated.

c. Orientation assignment

Both the magnitude and direction of Gaussian-smoothed image gradients at every pixel in the neighborhood area of a key point are computed. An orientation histogram is calculated (total N bins and each bin covers 360/N degrees) at each key point. The orientation of each pixel is weighted by its magnitude and inserted into the orientation histogram.

d. Descriptor

The local gradient data computed in the previous step, is also used to create key point descriptors. Key point descriptors usually use a set of 16 (4x4 grid) histograms, each with 8 orientation bins (covering 360 degree). Therefore, feature descriptor includes 128 dimensions.

After SIFT features of the same object scene in multiple views are extracted and stored into a database, features are matched against each other to find *k* nearest-neighbors for each feature.

Above feature detection methods are used to search distinctive image features. A non-maximal suppression mechanism is usually used to guarantee that no duplicate features are found in a local region.

Other local feature descriptors

SIFT performs very well in many applications, for example, in feature tracking and wide baseline camera matching. However, it has high computation expenses. Therefore, many different local feature descriptors, running in real-time or close to real-time, are introduced. These descriptors include Speeded Up Robust Features (SURF, Bay et al. 2008), Features from Accelerated Segment Test (FAST Rosten and Drummond 2006) and Daisy, which is an efficient dense descriptor (Tola et al 2010).

4.3 Advanced Topics (TO DO)

Here we discuss a few advanced techniques in stereo matching. They are mainly used in the correlation-based approach, but they can be applied to feature-based match algorithms as well.

- **Image filtering to handle illumination changes**
  - Image equalization
    - To make two images more similar in illumination
  - Laplacian filtering (2nd order derivative)
    - Use derivative rather than intensity (or original color)
- **Adaptive windows to deal with multiple disparities**
  - Adaptive Window Approach (Kanade and Okutomi)
    - Statistically adaptive technique which selects at each pixel the window size that minimizes the uncertainty in disparity estimates
    - **A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment**, *T. Kanade and M. Okutomi.* Proc. *1991 IEEE International Conference on Robotics and Automation*, Vol. 2, April, 1991, pp. 1088-1095
  - Multiple window algorithm (Fusiello, et al)
    - Use 9 windows instead of just one to compute the SSD measure
    - The point with the smallest SSD error among the 9 windows and various search locations is chosen as the best estimate for the given points

- A Fusiello, V. Roberto and E. Trucco, Efficient stereo with multiple windowing, IEEE CVPR pp858-863, 1997

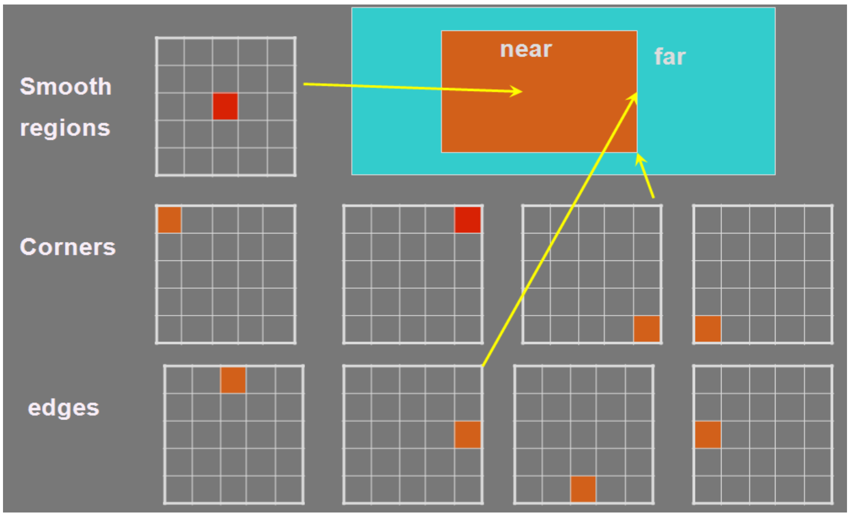


Figure 15. Multiple window approach

- Sub-pixel matching to improve accuracy
  - Find the peak in the correlation curves
- Self-consistency to reduce false matches esp. for occlusions
  - Check the consistency of matches from L to R and from R to L
- Multiple Resolution Approach
  - From coarse to fine for efficiency in searching correspondences
- Local warping to account for perspective distortion
  - Warp from one view to the other for a small patch given an initial estimation of the (planar) surface normal
- Multi-baseline Stereo
  - Improves both correspondences and 3D estimation by using more than two cameras (images)

5. 3D RECONSTRUCTION PROBLEM

So far we have dealt with the following two important issues:

(1) Finding correspondences using either correlation or feature based approaches.

(2) Defining epipolar geometry from at least 8 point correspondences.

Now we are going to discuss three cases of 3D reconstruction, depending on the amount of a prior knowledge of the stereo system.

(1) If both intrinsic and extrinsic parameters are known, we can solve the reconstruction problem unambiguously by triangulation.

(2) If only the intrinsic parameters are known, we can recover structure and extrinsic parameters up to an unknown scaling factor.

(3) If only image correspondences are known, we can achieve reconstruction only up to an unknown, global projective transformation (*optional – further reading)

5.1 Reconstruction by Triangulation

- Assumption and Problem
  - Under the assumption that both intrinsic and extrinsic parameters are known
  - Compute the 3-D location of a point from its projections, p_l and pr
- Solution
  - Triangulation: Two rays are known and their intersection can be computed
  - Problem: Two rays will not actually intersect in space due to errors in calibration, correspondences and pixelization
  - Solution: find a point in space with minimum distance from both rays

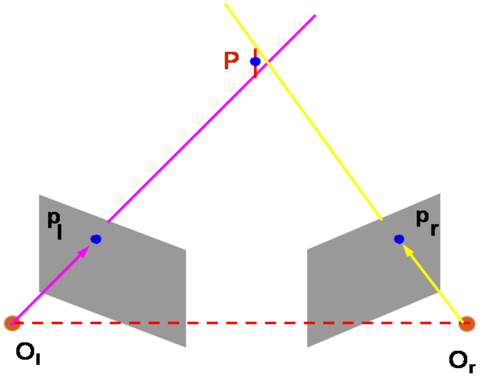


Figure 16. Reconstruction by triangulation

This reconstruction is solved by triangulation since we already know both the extrinsic and intrinsic parameters. Visually, we need to find the intersection of line O_lP_l and O_rP_r. The intersection is the 3D coordinate of corresponding projected points P_l and P_r. Because of the error in the location of corresponding points, the two rays will not intersect exactly in space and thus we have to approximate it by finding the closest midpoint between these two lines.

The point P_l can be represented on the line passing through O_l and P_l, A_0p_l in the left coordinate system, where A_0 is a real number, and p_l is the projected point of P in left image. The point P_r can be represented on the line passing through O_r and p_r, also in the left coordinate system, as $T + B_0R^Tp_r$, where B_0 is a real number, T is the translation matrix, R is the rotation matrix, and p_r is the projected point of P in right image.

If the two lines intersect in space, then we have $A_0P_l = T+ B_0R^TP_r$. If the two lines do not precisely intersect in space, we can find a line that is perpendicular to both lines, as $C_0(P_l \text{ x } R^TP_r)$. To solve for the midpoint, we need to solve a system of equation, as presented in [3]:

$$A_0P_l - B_0R^TP_r + C_0(P_l \text{ x } R^TP_r) = T$$

Then A_0, B_0, C_0 can be estimated. Suppose that the two intersections of this line with the two rays are given as:

$$P_1 = A_0P_l$$
$$P_2 = T + B_0R^TP_r$$

Therefore, the midpoint can be represented as

$$P_1 + c(P_2 - P_1)$$

With c = ½.

5.2 Reconstruction up to a Scale Factor

- Assumption and Problem Statement
  - Under the assumption that only intrinsic parameters and at least 8 point correspondences are given
  - Compute the 3-D location from their projections, p_l and p_r, as well as the extrinsic parameters
- Solution
  - Compute the essential matrix **E** from at least 8 correspondences
  - Estimate **T** (up to a scale and a sign) from **E (=RS)** using the orthogonal constraint of **R**, and then estimate **R**
    - End up with four different estimates of the pair (**T, R**)

- Reconstruct the depth of each point, and pick up the correct sign of **R** and **T**.
- Results: reconstructed 3D points (up to a common scale);
- The scale can be determined if distance of two points (in space) are known

5.3 Reconstruction up to a Projective Transformation (*)

- Assumption and Problem Statement
    - Under the assumption that only n (>=8) point correspondences are given
    - Compute the 3-D location from their projections, p_l and p_r
- Solution
    - Compute the fundamental matrix **F** from at least 8 point correspondences, and the two epipoles
    - Determine the projection matrices
        - Select five points (from correspondence pairs) as the projective basis
    - Compute the projective reconstruction
        - Unique up to the unknown projective transformation fixed by the choice of the five points

6. CONCLUDING REMARKS

This chapter discusses the following important topics:

- Fundamental concepts and problems of stereo
- Epipolar geometry and stereo rectification
- Estimation of fundamental matrix from 8 point pairs
- Correspondence problem and two techniques: correlation and feature based matching
- Reconstruct 3-D structure from image correspondences given
    - Fully calibrated
    - Partially calibration
    - Uncalibrated stereo cameras (*)

7. QUESTIONS AND PROJECTS

7.1 Questions

[Question 1]. Estimate the accuracy of the simple stereo system (Figure 7.4 in Trucco & Verri's book) assuming that the only source of noise is the localization of corresponding points in the two images. Discuss the dependence of the error in depth estimation as a function of the baseline width and the focal length.

Hint: Take the partial derivatives of Z with respect to x, T, and f, respectively.

[Question 2]. Prove that the fixation stereo does not change the fact that the depth resolution is inversely proportional to the square of depth.

[Question 3]. Prove that the epipolar line about p_r in the right image will pass through the location p_l if and only if R = I.

[Question 4]. Formulate the rectification rotation matrix $R_{rect,}$ and prove that the rotation between the two "rectified" cameras is I, and the translation has the form of (B, 0, 0), thus forming a stereo vision system with parallel optical axes.

7.2 Projects

Use an image pair (Image 1, Image 2) for the following exercises.

(1). Fundamental Matrix. - Design and implement a program that, given a stereo pair, determines at least eight pairs of point matches, then recovers the fundamental matrix and the location of the epipoles. Check the accuracy of the result by measuring the distance between the estimated epipolar lines and image points not used by the matrix estimation. Also, overlay the epipolar lines of control points and test points on one of the images (say Image 1- I already did this in the starting code below). Control points are the correspondences (matches) used in computing the fundamental matrix, and test points are those used to check the accuracy of the computation.

Hint: As a first step, you can pick up the matches of both the control points and the test points manually. You may use my Matlab code (FmatGUI.m) as a starting point - where I provided an interface to pick up point matches by mouse clicks. The epipolar lines should be (almost) parallel in this stereo pair. If not, something is wrong either with your code or the point matches. Make sure this is achieved before you move to the second step* - that is, to try to search for point matches automatically by your program. However, the second step is optional.

(2). Feature-based matching. - Design a stereo vision system to do "feature-based matching" and explain your algorithm in writing. The system should have a user interface that allows a user to select a point on the first image, say by a mouse click. The system should then find and highlight the corresponding point on the second image, say using a cross hair. Try to use the epipolar geometry derived from (1) in searching correspondences along epipolar lines.

Hint: You may use a similar interface as I did for question (1). You may use the point match searching algorithm in (1) (if you have done so), but this time you need to constrain your search windows along the epipolar lines.

(3) Discussions. Show your results on points with different properties like those in corners, edges, smooth regions, textured regions, and occluded regions that are visible only in one of the images. Discuss for each case, why your vision system succeeds or fails in finding the correct matches. Compare the performance of your system against a human user (e.g. yourself) who marks the corresponding matches on the second image by a mouse click.

REFERENCES

Digital Image Processing
Computer Graphics
Photogrammetry

[3] E.Trucco,. Introductory Techniques for 3-D Computer Vision. Upper Saddle River, NJ: Prentice Hall, 1998

J. Shi and C. Tomasi. "Good Features to Track," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 593 - 600, 1994.

C. Harris and M. Stephens. "A Combined Corner and Edge Detector," Fourth Alvey Vision Conference, pp. 147-51, 1988.

K. Mikolajczyk and C. Schmid, Scale and affine invariant interest point detectors, IJCV, vol. 1, no. 60, pp. 63 - 86, 2004.

D. Lowe, "Distinctive Image Features from Scale - Invariant Keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91 - 110, 2004.

H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346 - - 359, 2008

E. Rosten and T. Drummond, "Machine learning for high - speed corner detection," in Proceedings of the European Conference on Computer Vision, pp. 430 – 443, 2006

E. Tola, V. Lepetit, P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. PAMI, May 2010,