Part 2 – Uncalibrated Stereo Object Size Estimation

(Plain ASCII version for RStudio)

1. Setup and Notation

We assume uncalibrated stereo.

We have two unknown cameras (left and right).
A 3D point is written as:

X = (X, Y, Z, 1)

Its projection in the left image is:

x = (u, v, 1)

Its projection in the right image is:

x_prime = (u_prime, v_prime, 1)

The projection equations are:

x is proportional to P * X
x_prime is proportional to P_prime * X

P and P_prime are unknown 3x4 projection matrices.
We do NOT know the intrinsic matrix K, focal length f, or the baseline B.
Therefore the formula Z = f * B / disparity cannot be used.
Instead, we use epipolar geometry.

2. Epipolar Geometry and the Fundamental Matrix F

For any corresponding pair x <-> x_prime, the epipolar constraint must hold:

x_prime_transpose * F * x = 0

This equation defines the fundamental matrix F.

2.1 Conceptual Explanation

The two camera centers and any 3D point define an epipolar plane.
This plane intersects each image plane in an epipolar line.

In the right image, the epipolar line is:

l_prime = F * x

The corresponding point x_prime must lie on this line.
Therefore the constraint:

x_prime_transpose * F * x = 0

must hold for every correspondence.

2.2 Estimating F (8 Point Algorithm)

Given N >= 8 correspondences (u_i, v_i) <-> (u_prime_i, v_prime_i):

Each correspondence produces a linear equation in the entries of F.
The equation has the form:

u_prime_i * u_i * f11
+ u_prime_i * v_i * f12
+ u_prime_i * f13
+ v_prime_i * u_i * f21
+ v_prime_i * v_i * f22
+ v_prime_i * f23
+ u_i * f31
+ v_i * f32
+ f33 = 0

Stacking all equations produces a system A * f = 0.
Solve using singular value decomposition.
Reshape f into a 3x3 matrix to obtain F.
Force rank(F) = 2 by zeroing the smallest singular value.

3. Constructing Uncalibrated Camera Matrices P and P_prime

Once F is known, we can construct a canonical pair of projection matrices.

Step 1: Set the left camera to:
P = [I | 0]

Step 2: Compute the right epipole e_prime such that:
F_transpose * e_prime = 0

Step 3: Form the skew symmetric matrix [e_prime]_x.

Step 4: Define the right camera as:
P_prime = ([e_prime]_x * F | e_prime)

This pair of cameras is valid for uncalibrated stereo.

Note:
The reconstruction is defined only up to a 4x4 projective matrix H.
Applying the transformation:

P -> P * H
P_prime -> P_prime * H
X -> H_inverse * X

does not change the image projections.

4. Triangulating 3D Points

For each correspondence x <-> x_prime:

x is proportional to P * X
x_prime is proportional to P_prime * X

Use the cross product equations:

x cross (P * X) = 0
x_prime cross (P_prime * X) = 0

Together they produce 4 linear equations in the 4 components of X.
This can be written as:

A * X = 0

Solve using singular value decomposition.
This gives the homogeneous 3D point X, up to projective scale.

Repeat this for all correspondences to reconstruct the entire object.

5. Converting Projective 3D Into Real Distances

Uncalibrated stereo gives the correct shape but not the correct scale.
To measure real distances, we use a reference length.

Step 1: Choose two reconstructed 3D points Xa and Xb whose real distance is known.
Call this real distance d_real.

Step 2: Compute the reconstructed distance:

d_recon = sqrt((Xa.x - Xb.x)^2
+ (Xa.y - Xb.y)^2
+ (Xa.z - Xb.z)^2)

Step 3: Compute the global scale:

s = d_real / d_recon

Step 4: For any pair Xi and Xj on the object:

object_distance = s * sqrt((Xi.x - Xj.x)^2
+ (Xi.y - Xj.y)^2
+ (Xi.z - Xj.z)^2)

This produces real-world measurements.

6. Final Procedure Summary

1. Collect corresponding points between the two images, including one known reference length.

2. Estimate the fundamental matrix F using the 8 point algorithm.

3. Construct canonical cameras P and P_prime.

4. Triangulate 3D points.

5. Compute global scale using the known reference.

6. Measure object dimensions by computing scaled 3D distances.