(20878) Computer Vision & Image Processing - Homework 1

Beatrice Citterio, Clara Montemurro, Martina Serandrei
Group name: Girls' POV

1 Single view metrology

In this section, we estimate the height of a person in an image by exploiting geometric invariants, vanishing points, the horizon line, and the projective cross-ratio, using a reference object of known height.

1.1 Problem Setup and Image Annotation

The image analyzed depicts two individuals standing on a common horizontal ground plane in front of a building. The person in the foreground (Martina) is selected as the reference subject, while the person in the background (Clara) is the target subject whose height is to be estimated (Figure 1).
The actual heights are:

- Reference height: $h_{\text{ref}} = 170\,\text{cm}$;
- Target height (ground truth): $h_{\text{target}} = 162\,\text{cm}$.

We manually annotate the heads $\{H_{\text{ref}}, H_{\text{targ}}\}$ and feet $\{F_{\text{ref}}, F_{\text{targ}}\}$ points of both subjects in homogeneous coordinates, resulting in two vertical line segments corresponding to their apparent image heights.

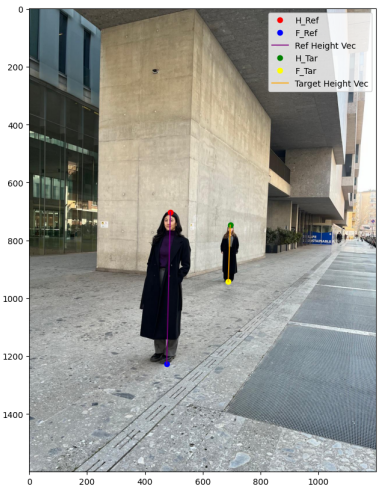


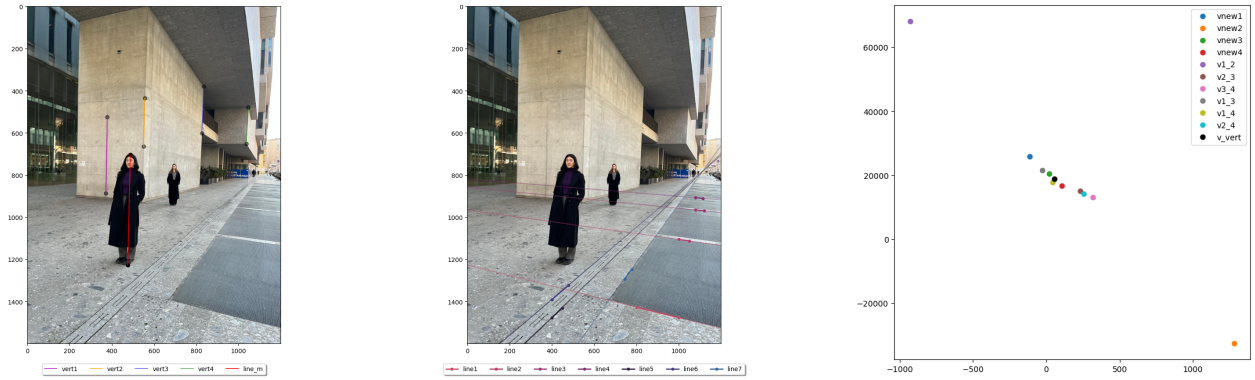
Figure 1: Head-Feet segments of the subjects

1.2 Vanishing Point Estimation and Horizon Line

To enable metric reasoning within the image, we compute three vanishing points: one vertical (v_{vert}) and two horizontal ($v_{\text{left}}, v_{\text{right}}$). These points are obtained from sets of real-world parallel lines, as the points in the image where parallel lines intersect.
The procedure consists of:

- **Selecting parallel line pairs** in the image (e.g., building edges, floor tiles);
- **Computing line equations** via the cross product of two annotated points;

- **Computing vanishing point estimates** as intersection points of line pairs;
- **Refining the estimate using SVD**: stacking the line vectors into a matrix and extracting the right singular vector corresponding to the smallest singular value. Implementing the SVD method maximizes consistency across the different estimates and minimizes the impact of noise in the data.



(a) Parallel vertical lines (b) Parallel horizontal lines (c) Refined vertical vanishing point

Figure 2: Vanishing Points Estimation

The procedure is applied for both vertical and horizontal lines.
To improve robustness, the vertical vanishing point is estimated using both architectural vertical lines and the reference height vector, assumed to be perpendicular to the ground. Similarly, horizontal vanishing points are estimated using lines lying on the ground plane.
The horizon line is then defined by:

$$\ell_{\text{horiz}} = v_{\text{left}} \otimes v_{\text{right}} \tag{1}$$

This line corresponds to all points in the image at eye level and is crucial for consistent height estimation.

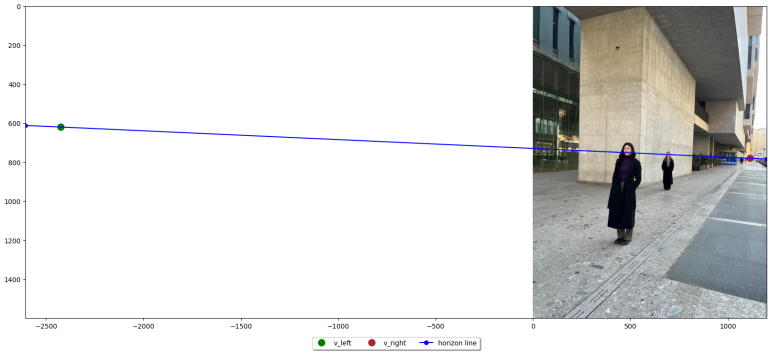


Figure 3: Horizontal vanishing points

1.3 Height Estimation via Cross-Ratio

We now estimate the height of the target subject by leveraging the projective cross-ratio, an invariant under perspective projection. First, we define the vertical line passing through the key points of the reference subject as $\ell_{\text{ref}} = H_{\text{ref}} \otimes F_{\text{ref}}$.
Then, we align the two subjects geometrically via projective constructions:

- Construct the *ground line*: $\ell_{\text{ground}} = F_{\text{ref}} \otimes F_{\text{targ}}$;
- Compute the intersection point of this line with the horizon: $U = \ell_{\text{ground}} \otimes \ell_{\text{horiz}}$;
- Construct the *projection line* $\ell_{\text{proj}} = U \otimes H_{\text{targ}}$;

- Compute the intersection point $I = \ell_{\text{proj}} \otimes \ell_{\text{ref}}$.

The previous steps can be visualized in Figure 3.

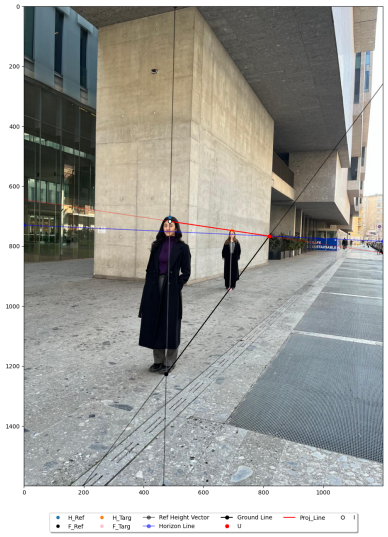


Figure 4: Points to estimate the target's height

These constructions provide four collinear points along ℓ_{ref}: $H_{\text{ref}}, I, F_{\text{ref}}, v_{\text{vert}}$. Their configuration allows the application of the cross-ratio formula:

$$CR = \frac{\|I - F_{\text{ref}}\| \cdot \|v_{\text{vert}} - H_{\text{ref}}\|}{\|H_{\text{ref}} - F_{\text{ref}}\| \cdot \|v_{\text{vert}} - I\|} \tag{2}$$

Leveraging the invariant property of the cross-ratio for projective transformations, we deduce the following relation:

$$\frac{\hat{h_{\text{targ}}}}{h_{\text{ref}}} = \frac{\|I - F_{\text{ref}}\|\|v_\infty - H_{\text{ref}}\|}{\|H_{\text{ref}} - F_{\text{ref}}\|\|v_\infty - I\|}$$

which yields the estimated height:

$$\hat{h}_{\text{targ}} = h_{\text{ref}} \cdot CR$$

The estimation we obtained for the target's height is $\hat{h_{\text{targ}}} = 165.53$cm that, when validated against the ground value, results in an estimation error of approximately 3.5 cm.

2 The Eight-Points Algorithm

The goal of this section is to implement the eight-points algorithm and show that it works on the two images shown in Figure 5.

2.1 Background of two View Geometry

Consider two images I_1, I_2 with two corresponding camera centers C_1 and C_2. Let M be a 3D point. **Corresponding points** are pairs of points (x_i, x_i') s.t. $x_i \in I_1 \subset \mathbb{P}^2$, $x_i' \in I_2 \subset \mathbb{P}^2$ and

$$\begin{cases} \lambda_1 x_i = P_1 M \\ \lambda_2 x_i' = P_2 M \end{cases}$$

i.e. x_i, x_i' are the projections via P_1 and P_2 respectively of the very same 3D point M.

We denote **baseline** as the line passing through the camera centers, **epipolar plane** as the plane containing M and the baseline, **epipoles** as the intersections e_1, e_2 between image planes and the baseline and, finally

epipolar lines as the lines ℓ_1, ℓ_2 which are the intersections of the epipolar plane and the image plane in the two images. Epipolar lines in the first image (left) can be computed as

$$\ell = F^T \; x_i'$$

while epipolar lines in the second image (right) are computed as

$$\ell = F \; x_i$$

where F is the **fundamental matrix**, defined as

$$F = [e_2]_\times P_{2,\ 1:3} P_{1,\ 1:3}^{-1}$$

In other words, it is the 3×3, rank 2 matrix that satisfies $(x_i')^T F x_i = 0$ for corresponding points (x_i, x_i').

The **eight-points algorithm** is the procedure which allows us to estimate the fundamental matrix F given eight (or more) correspondences. For the details of the steps see Section 2.3.

2.2 Data Loading and Correspondences

The first step is to find correspondences between images, which will be denoted as $\{x_i, x_i'\}_{i=1}^n$. This could have been done manually or with automated feature detection methods, such as SIFT. However, to avoid noise and misclassifications, we decided to manually annotate the correspondences. We decided to only choose 10 points for each image as the algorithm performs well on these alone. However, the same script can be used on more correspondences, as it works on any two sets of points with length n larger than 8.
Once we have the correspondences, shown in Figure 5, we can implement the eight-points algorithm.

Figure 5: Plots of the two images and correspondences

2.3 Algorithm

The goal of the algorithm is to estimate the fundamental matrix F. To do so, the following steps are followed:

1. **preconditioning of points** - the first step consists in normalizing the points, i.e. shift them so that they are centered around the origin and the average distance from the origin is $\sqrt{2}$. This is done using the auxiliary functions `norm_matrix()` (which computes the normalization matrix of a given set of points, i.e. the transformation which must be applied) and `convert_to_homo()` (which makes the coordinates homogeneous by adding a third component).

2. **build linear system** - the second step consists in building the matrix A which is defined such that its ith row corresponds to the cross product between $(x_i')^T$ and x_i^T, i.e.

   $$A_i = [(x_i')^T \otimes x_i^T]$$

   for $i = 1, ..., n$

3. **SVD** - now, we can solve the linear system $Az = 0$, and we do so using singular value decomposition. In other words, we decompose A as $A = U\Sigma V^T$, we take the vector $f \in \mathbb{R}^9$ in V^T corresponding to the smallest singular value, and we reshape it to obtain a 3×3 matrix.

4. **impose rank deficiency** - as we know, the fundamental matrix F must me rank deficient, i.e. $\text{rank}(F) = 2$. To impose so, we again compute the SVD decomposition of the matrix f we obtained, i.e. $f = \tilde{U}\tilde{\Sigma}\tilde{V}^T$. Then we set the smallest singular value to 0, obtaining $\tilde{\Sigma}'$ and use this to compute F as
$$F = \tilde{U}\tilde{\Sigma}'\tilde{V}^T$$

5. **denormalize and rescale** - finally, we de-normalize F by premultiplying it by the transpose of the normalization matrix of the second set of points and post multiplying it by the other normalization matrix. To conclude, we rescale it using its Frobenius norm.

2.4 Epipolar Lines

Now we compute the equations of the epipolar lines as written in Section 2.1. The plot of epipolar lines can be found in Figure 6.

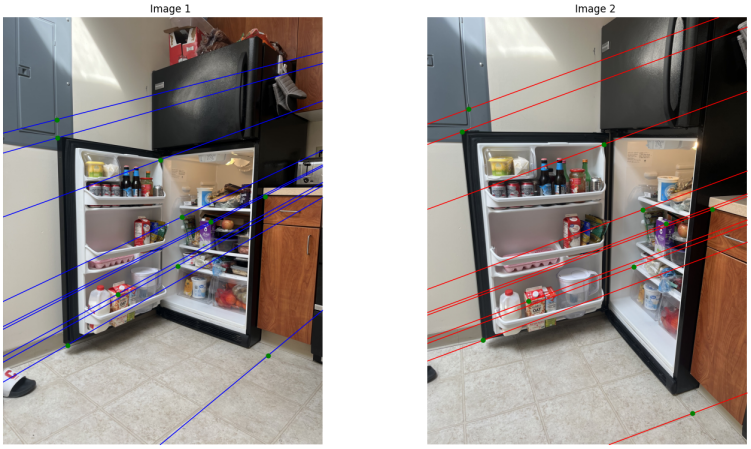


Figure 6: Plot of Epipolar Lines

The last step is to use a geometric error to see how good our estimate is. To do so, we compute the epipolar distance between points and their epipolar lines.
For each index $i = 1, ..., n$, where n is the number of correspondences, we compute the epipolar line in image 1 corresponding to x_i' and the epipolar line in image 2 corresponding to x_i. Then, we compute the normalized distance of the point in image 1 with the epipolar line in image 1 and same for image 2. The distance from x_i to the epipolar line in image 1 corresponding to x_i' is given by
$$\frac{|x_i^T F^T x_i'|}{\sqrt{(a^2 + b^2)}}$$
where a, b are the parameters of the line, i.e. the line is defined as $[a, b, c]$.

Similarly, the distance from x_j' to the epipolar line in image 2 corresponding to x_j is given by
$$\frac{|(x_j')^T F x_j|}{\sqrt{(a^2 + b^2)}}$$
where, again, a, b are the parameters of the line.
For each index i we then compute the mean of these two distances and this will be the epipolar error of index i.

To assess whether our results are good or not, we compute the mean epipolar error across all indices i. When `FACTOR = 0.75` (i.e. the image is scaled to 75% of its original dimension), the mean error is 1.504677 pixels.