20878 – Computer Vision and Image Processing

Homework (Geometry)

MSc in Artificial Intelligence – Bocconi University

"Rectory" Group

Alessandro Bellini 3169373
Davide Beltrame 3306906
Giacomo Cirò 3160499

Spring 2025

Figure 1: The image we used in the second part of the homework to run the eight-point algorithm.

1 Single View Metrology

Introduction We consider an image of two people standing on the same ground plane (Figure 2), annotated[1] with:

- Two pairs of parallel lines in the scene.
- Two points identifying a segment of known length, i.e. the reference height.
- Two points identifying a segment of unknown length, i.e. the height to be estimated.

Figure 2: In blue and red, the points identifying the two pairs of parallel lines. In yellow, the points representing the height to be estimated (Jack's). In purple, the points representing the reference height (Dave's). Note that in the image we use the 0-based notation (points $j,1$ and $j,2$ identify line l_{j+1})

Figure 3: In blue and red, the lines retrieved using the annotated points and the cross-product property. In yellow and purple, the segment representing the height of the person. The purple segment's length is known, the yellow segment's length is the one we estimate.

Notation In this section, we use the following notation:

- *Person 1* (Dave): the person whose height is known and we use as reference.
- *Person 2* (Jack): the person whose height is unknown and we want to estimate.
- h_i, f_i: the points identifying Person i's height, for $i = 1, 2$.
- $p_{j,1}$, $p_{j,2}$: the points identifying the j-th parallel line annotated in the scene, for $j = 1, 2, 3, 4$.

Background Our methodology heavily relies on a key property of projective geometry, which involves homogeneous coordinates and the cross product. Let $x, y, v, w \in \mathbb{P}^2$ be four points in the projective plane and let $\times$ denote the cross product in $\mathbb{R}^3$. Then the following holds:

- $x \times y$ are the homogeneous coordinates in the dual projective plane of the line passing through x and y.
- $(x \times y) \times (v \times w)$ are the homogeneous coordinates in the projective plane of the intersection of the line passing through x and y with the line passing through v and w.

[1]We developed an ad-hoc annotator file to obtain pixel-precise manual point coordinates.

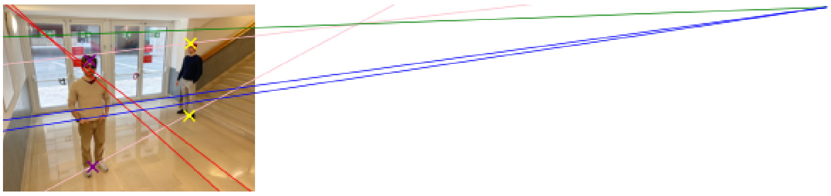

Figure 4: In green, the retrieved horizon. The left and right vanishing points are represented by the intersection of the horizon with the red lines and blue lines respectively.

Methodology To estimate the height, we use the annotations to identify the pairs of parallel lines and their intersection points. We compute the horizon as the line passing through these intersections. We find the intersection of the line through the feet of the two people with the horizon. From this point, we draw a line to project the height of one person onto the other. Finally, we measure their ratio in the image reference system and use a proportion to retrieve the original height. More in details:

1. Extend the 2D coordinates of the points in the image reference system by setting the third coordinate to 1 to get the homogeneous coordinates and identify the parallel lines l_j (Figure 3):

$$l_j = p_{j,1} \times p_{j,2} \quad \text{for } j = 1, 2, 3, 4$$

2. Identify the vanishing points and the horizon (Figure 4):

$$\begin{aligned} v_{left} &= l_1 \times l_2 \\ v_{right} &= l_3 \times l_4 \\ l_\infty &= v_{left} \times v_{right} \end{aligned}$$

3. Identify the line passing through the feet f_1 and f_2 of the two people (Figure 5:):

$$l_{feet} = f_1 \times f_2$$

4. Find the point at infinity where l_{feet} intersects with the horizon (Figure 5:):

$$p_\infty = l_{feet} \times l_\infty$$

5. Project the Person 2's top point t_2 onto the line l_1 where Person 1's height segment lies (Figure 5):

$$\begin{aligned} l_1 &= t_1 \times f_1 \\ l_{heads} &= p_\infty \times t_2 \\ t_2' &= l_{heads} \times l_1 \end{aligned}$$

6. Normalize the projected point to get the coordinates in the image reference system:

$$t_2' = \frac{t_2'}{(t_2')_3}$$

7. Measure Person 1's height h_1 and Person 2's projected height h_2' in the picture:

$$\begin{aligned} h_1 &= \|t_1 - f_1\|_2 \\ h_2' &= \|t_2' - f_1\|_2 \end{aligned}$$

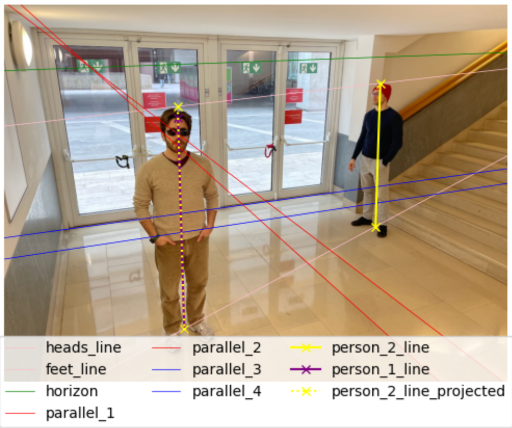

Figure 5: The final result. The purple line (Person 1's height) is known to be 175 cm in reality. The yellow line (Person 2's height) is estimated to be 184.10 cm (which in reality is 184 cm). The dashed yellow line represents the projection of the yellow line onto the purple line.

8. Use a proportion to estimate Person 2's original height $\hat{h}_2^*$ given Person 1's annotated height h_1^* (Figure 5):

$$h_1^* : \hat{h}_2^* = h_1 : h_2' \implies \hat{h}_2^* = \frac{h_2' \cdot h_1^*}{h_1}$$

Implementation We implement our solution in *Python*, using *NumPy* for matrix operations and *PIL* for image handling.

Results We test our approach on an image where the actual height of the second person was known to be 184 cm. Using a reference person with a height of 175 cm, our algorithm estimates the second person's height to be 184.10 cm.

Discussion On this particular image, the final result was extremely precise. However, we notice that this methodology is very sensitive to small shifts in the annotations, and on different test images it performs slightly worse. For example, we consider an image where the left vanishing point is extremely close to one person's top reference point, which greatly magnifies annotation errors and yields suboptimal results. Additionally, we notice that using two pairs of parallel lines as references in the scene such that two non-parallel lines intersect at an annotated point (Point 2, 2 in Figure 2) significantly improves the estimation accuracy.

2 The Eight Point Algorithm

Introduction We implement the eight-point algorithm to estimate the fundamental matrix F given two images representing two views of the same scene, which allows us to relate points in one image to epipolar lines in the other. Given eight points correspondences, we derive a linear system where the unknown are the entries of the fundamental matrix by transforming the epipolar constraints using the *Kronecker product trick*. We normalize the points using *Hartley normalization* and we approximate a solution to the linear system using SVD. The entire process is repeated multiple times according to the *RANSAC* framework for robust estimation.

Notation In this section, we use the following notation:

- $\mathbf{x}_i$, $\mathbf{x}'_i$: corresponding points in the first and second image in homogeneous coordinates.
- F: the fundamental matrix, to be estimated.
- l_i, l'_i: epipolar lines in the first and second image corresponding to $\mathbf{x}'_i$ and $\mathbf{x}_i$.

Background The fundamental matrix F is the unique 3×3 rank-2 homogeneous matrix that satisfies the epipolar constraint for any pair of corresponding points $\mathbf{x}_i$ and $\mathbf{x}'_i$:

$$\mathbf{x}'^T_i F \mathbf{x}_i = 0 \quad \text{s.t. } Rank(F) = 2$$

For a point $\mathbf{x}_i$ ($\mathbf{x}'_i$) in the first (second) image, the corresponding epipolar line l'_i (l_i) in the second (first) image can be computed as:

$$l'_i = F\mathbf{x}_i \quad (l_i = F^T \mathbf{x}'_i)$$

Given a point $\mathbf{x}_i$ and its corresponding point $\mathbf{x}'_i$, we defune the *geometric error* $E(\mathbf{x}_i)$ as:

$$E(\mathbf{x}) = \text{distance}(\mathbf{x}_i, F^T \mathbf{x}'_i)$$

Where $\text{distance}(p, l)$ measures the euclidean distance of the point p from the line l.

Methodology Our implementation is illustrated in the following steps:

1. We obtain 30 point correspondences $\{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^{30}$ between two images using manual annotation[2].
2. We use *Hartley normalization* to improve numerical stability of the following computations by transforming the points such that their mean is zero and the average distance from the origin is $\sqrt{2}$. We compute and store the matrices T_1, T_2 performing this linear transformation on the two set of points:

$$\tilde{\mathbf{x}}_i = T_1 \mathbf{x}_i$$
$$\tilde{\mathbf{x}}_i = T_2 \mathbf{x}_i$$

   These matrices will be used later to de-normalize the fundamental matrix estimated using the normalized points.
3. We use the *Eight points algorithm* to estimate the fundamental matrix F:

[2] We developed another ad-hoc annotator file to obtain pixel-precise manual point correspondences. We also attempted to use *SIFT* for automatic feature detection and matching, but it did not produce satisfactory results in our experiments, so we stick with manual annotation.

(a) Randomly sample a subset of 8 correspondences

(b) Exploit the *Kronecker product trick* and the epipolar constraints to set up a system of 8 linear equations in 9 unknowns (the entries of the matrix F).

$$x_i' x_i f_{11} + x_i' y_i f_{12} + x_i' f_{13} + y_i' x_i f_{21} + y_i' y_i f_{22} + y_i' f_{23} + x_i f_{31} + y_i f_{32} + f_{33} = 0 \quad \text{for } i = 1, ..., 8$$

Where $\tilde{\mathbf{x}}_i = (x_i, y_i, 1), \tilde{\mathbf{x}}_i' = (x_i', y_i', 1)$ and f_{ii} are the entries of the matrix F.

(c) Solve the system via SVD and reshape the solution into a 3×3 matrix.

(d) Use SVD again to get the best rank-2 approximation of this matrix by setting the last singular value to 0. Call this reconstructed matrix $\tilde{F}$.

(e) De-normalize F such that it operates on points that are not Hartley normalized:

$$F = T_2^\top \tilde{F} T_1$$

(f) Scale it such that $F_{2,2} = 1$ (common convention).

4. To make this process robust, we repeat it N times according to the *RANSAC* framework. At each iteration, we compute the geometric error $E(x_i)$ and keep track of the set of inliers, i.e., those points x_i such that $E(x_i) < \epsilon$. Finally, we use the largest set of inliers to estimate the final matrix F.

Implementation We implement our solution in *Python*, using *NumPy* for matrix operations, *OpenCV* for image processing and *Matplotlib* for visualization.

Figure 6: Epipolar lines for the two views. Each point in one image corresponds to a line in the other image, and all these lines should intersect at the epipole (which is oftentimes outside the visible image).

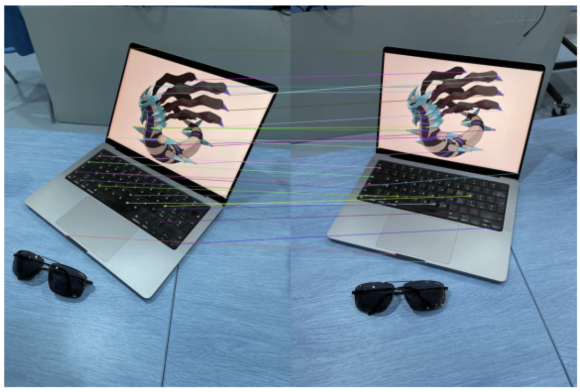

Figure 7: Manually annotated point correspondences between the two images. The lines connect matching points across the two views.

Results We test our implementation on a pair of images taken from slightly different viewpoints (the ones in Figure 6 and 7). Using *RANSAC* with $N = 1000$ and $\epsilon = 0.2$, we achieved a mean geometric error of 0.058 pixels (averaged over all points and over 100 trials, with a standard deviation of 0.019 and a max error 0.140). The algorithm successfully identified 14 inliers out of the 30 annotated correspondences, resulting in an inlier ratio of 47%.

Discussion Our implementation of the eight-point algorithm with normalization and RANSAC proved effective in estimating the fundamental matrix and visualizing the epipolar geometry. We observed that normalization significantly improved the numerical stability of the algorithm, while RANSAC successfully handled outliers in the point correspondences.