Image Analysis and Computer Vision - Homework

Christian Rossi

Academic Year 2024-2025

Contents

CHAPTER 1

Theory

1.1 Vanishing line

The objective of this task is to determine the vanishing line, denoted as $\mathbf{l}'_\infty$ of the horizontal plane. The horizontal plane is formed with the width and depth of the object.

Figure 1.1: Original image

The lines on the width and the depth of the object from now on will be called $\mathbf{l}_i$ and $\mathbf{m}_j$, respectively.

1.1.1 Vanishing points

To determine the vanishing points, we need to identify the intersection of two lines that are parallel in the real world but appear to converge in the image. This can be done thanks to the following theorem:

Theorem 1.1.1. *The image of a set of parallel lines $\mathbf{n}_i$ is a set of concurrent lines $\mathbf{nl}'_i$ that intersect at a common point $\mathbf{p}'$, referred to as the vanishing point of the direction of lines $\mathbf{n}_i$.*

For both the width and depth, we select two pairs of points that lie along the same line. To derive the equation of the line passing through two points, we compute the cross product in the 2D plane. In general, given two points $\mathbf{x}_1$ and $\mathbf{x}_2$, the line can be expressed as:

$$\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$$

By performing this operation for four pairs of points, we obtain two lines for each dimension:

$$\mathbf{lh}_1 \qquad \mathbf{lh}_2 \qquad \mathbf{lm}_1 \qquad \mathbf{lm}_2$$

The vanishing point for each dimension is found by intersecting the respective lines. This can be done by taking the cross product of the two lines:

$$\mathbf{p}_l = \mathbf{lh}_1 \times \mathbf{lh}_2$$

$$\mathbf{p}_m = \mathbf{lm}_1 \times \mathbf{lm}_2$$

In the given image, we ha found the following vanishing points:

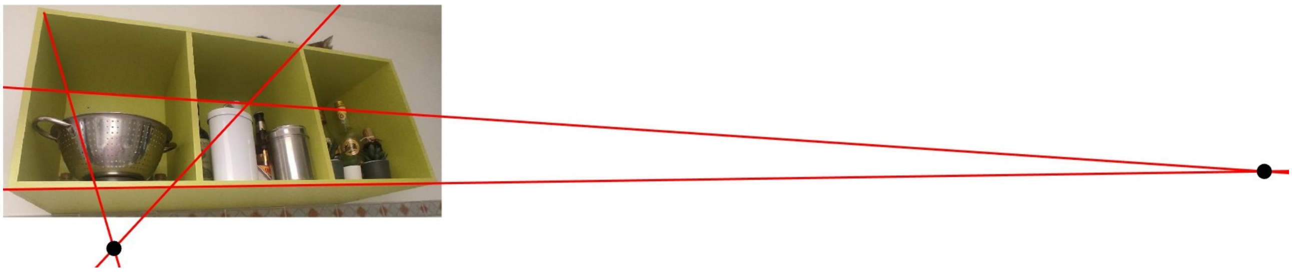

Figure 1.2: Vanishing points for depth and width

The coordinates of the vanishing points are given by:

$$\mathbf{p}_l = \begin{bmatrix} x_l \\ y_l \\ w \end{bmatrix} \qquad \mathbf{p}_m = \begin{bmatrix} x_m \\ y_m \\ w \end{bmatrix}$$

1.1.2 Vanishing line

The vanishing line $\mathbf{l}'_\infty$ is defined by the connection between the two vanishing points, $\mathbf{p}_l$ and $\mathbf{p}_m$. This is achieved by computing the cross product of the two points:

$$\mathbf{l}'_\infty = \mathbf{p}_m^T \times \mathbf{p}_l^T = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

In the given image, we ha found the following vanishing line:

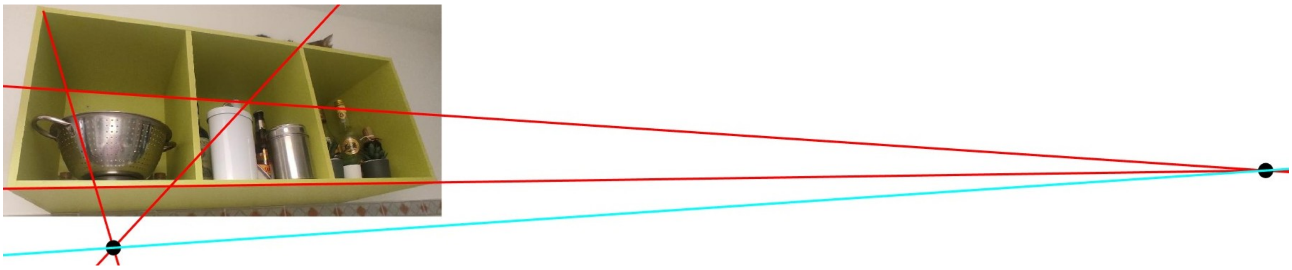

Figure 1.3: Vanishing line for depth and the width dimension

1.2 Rectification

The objective of this section is to establish a rectification mapping for the horizontal plane analyzed earlier and subsequently compute the object's depth. This process employs a stratified approach, beginning with Euclidean rectification followed by metric rectification.

1.2.1 Euclidean rectification

From geometric theory, the line at infinity identified in the previous step must be mapped to the real line at infinity in the scene. Specifically, the line $\mathbf{l}'_\infty$ in the image must be transformed back to $\mathbf{l}_\infty$:

$$\mathbf{l}'_\infty = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \xrightarrow{\text{mapped back to}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{l}_\infty$$

To achieve this, we construct a homography matrix that aligns the line $\mathbf{l}'_\infty$ with $\mathbf{l}_\infty$:

$$\mathbf{H}_{\text{rect}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & c \end{bmatrix}$$

By applying this transformation matrix to all pixels in the image, we obtain a rectified version of the scene. The resulting image is shown below:

Figure 1.4: Euclidean rectification

1.2.2 Metric rectification

Using a stratified approach, we proceed with the metric rectification step. This involves imposing orthogonality constraints to achieve a full metric reconstruction of the image.

We start by selecting two pairs of orthogonal lines: one pair along the lower side of the object and another pair intersecting near the top (ensuring the pairs are independent).

From the selected pairs of lines $\mathbf{l}$ and $\mathbf{m}$, we compute the orthogonality conditions and populate a matrix $\mathbf{A}$. Each row of $\mathbf{A}$ corresponds to the constraints imposed by one pair of lines, computed as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{l}_x \cdot \mathbf{m}_x & \mathbf{l}_x \cdot \mathbf{m}_y + \mathbf{l}_y \cdot \mathbf{m}_x & \mathbf{l}_y \cdot \mathbf{m}_y \\ \cdots & \cdots & \cdots \end{bmatrix}$$

After constructing $\mathbf{A}$ using the selected two pairs of lines, we apply Singular Value Decomposition (SVD) to it. The last row of the resulting $\mathbf{V}$ matrix contains the coefficients for the conic matrix $\mathbf{S}$, which is expressed as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{V}(1,3) & \mathbf{V}(2,3) \\ \mathbf{V}(2,3) & \mathbf{V}(3,3) \end{bmatrix}$$

To compute the homography matrix, we first find the transformation matrix $\mathbf{G}$, which is computed as

$$\mathbf{A} = \mathbf{U}\sqrt{\mathbf{D}}\mathbf{V}^T$$

Here, $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ are obtained by performing Singular Value Decomposition on $\mathbf{S}$.

The final metric rectification homography matrix is then given by:

$$\mathbf{H}_{\text{metric}} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$$

Applying $\mathbf{H}_{\text{metric}}$ to the rectified image transforms it into a fully metric rectified image. The resulting image is shown below:

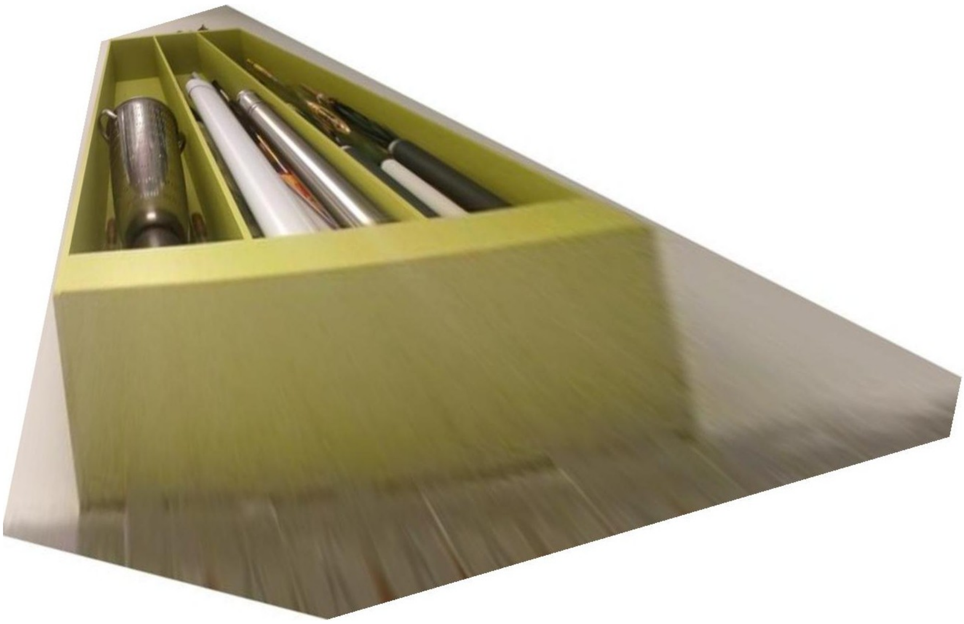

Figure 1.5: Metric rectification

1.2.3 Depth computation

The final step is to compute the object's depth using the metric rectified image. Assuming the object's width is known and unitary, we can measure both the width and depth directly on the rectified image using Euclidean distances between corresponding points.

The real-world depth is calculated as:

$$\text{real depth} = \frac{\text{depth}}{\text{width}} \cdot 1$$

This computation provides the actual depth of the object in proportion to its known width.

1.3 Calibration matrix

The goal of this section is to compute the calibration matrix $\mathbf{K}$, which encodes the intrinsic parameters of the camera. The calibration matrix is given by:

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Here:

- f_x and f_y are the effective focal lengths along the x and y-axes, respectively.
- u_0 and v_0 are the coordinates of the principal point (the optical center in the image).
- s is the skew parameter, which describes the angle between the image axes. By assumption, we are using a zero-skew camera, so $s = 0$.

1.3.1 Image of the Absolute Conic

To estimate the calibration matrix, we leverage the Image of the Absolute Conic (IAC), represented by the symmetric matrix:

$$\boldsymbol{\omega} = \begin{bmatrix} a & 0 & b \\ 0 & 1 & c \\ b & c & d \end{bmatrix}$$

The IAC encodes intrinsic camera parameters and can be used to recover the elements of $\mathbf{K}$.

To determine $\boldsymbol{\omega}$, we impose geometric constraints based on vanishing points and the line at infinity. We will use the following elements fo the contraints:

1. *Vertical vanishing point*: $\mathbf{ph}$.
2. *Homography matrix*: $\mathbf{H} = (\mathbf{H}_{\text{metric}}\mathbf{H}_{\text{rect}})^{-1}$, which combines Euclidean and metric transformations.

The constraints imposed on $\boldsymbol{\omega}$ are as follows:

$$\begin{cases} \mathbf{ph}^T\boldsymbol{\omega}\mathbf{h}_1 = 0 \\ \mathbf{ph}^T\boldsymbol{\omega}\mathbf{h}_2 = 0 \\ \mathbf{h}_1^T\boldsymbol{\omega}\mathbf{h}_2 = 0 \\ \mathbf{h}_1^T\boldsymbol{\omega}\mathbf{h}_1 - \mathbf{h}_2^T\boldsymbol{\omega}\mathbf{h}_2 = 0 \end{cases}$$

Here $\mathbf{h}_1$ and $\mathbf{h}_2$ are the first and second columns of the homography matrix $\mathbf{H}$, respectively.

The constraints form a linear system of equations that can be solved to recover the coefficients a, b, c, and d of the IAC matrix $\boldsymbol{\omega}$. Once $\boldsymbol{\omega}$ is determined, the intrinsic parameters of $\mathbf{K}$ can be computed.

1.3.2 Intrinsic parameters estimation

The intrinsic parameters are computed as follows:

$$\begin{cases} f_x = \sqrt{d - \alpha^2 \cdot u_0^2 - v_0^2} \\ f_y = \frac{f_y}{\alpha} \\ u_0 = -\frac{b}{\alpha^2} \\ v_0 = -c \end{cases}$$

Here, $\alpha = \sqrt{a}$.

1.4 Vertical side reconstruction

1.5 Circumference estimation

1.6 Localization

CHAPTER 2

MATLAB

2.1 Features detection

2.2 Theory implementation

2.3 3D model recovery