


FACULTY OF MATHEMATICS,
PHYSICS AND INFORMATICS
Comenius University
Bratislava

3D Vision

Leveraging Binocular Vision and Deep Learning for 3D Scene Understanding with NICO

Ing. Viktor Kocur, PhD.

11.6.2024

Contents

Homogeneous Coordinates

Homogeneous coordinates can be used to represent points and vectors. In 2D, a point with homogeneous coordinates (x, y, w) represents the point $(x/w, y/w)$ in Cartesian coordinates.

- Points with $w \neq 0$ represent finite points in the Euclidean plane. In this case, the coordinates (x, y, w) are proportional to the Cartesian coordinates $(x/w, y/w)$ of the corresponding point in 2D space.
- Points with $w = 0$ represent points at infinity. In this case, the coordinates $(x, y, 0)$ represent the direction of the line that passes through the point at infinity.
- The triple $(0, 0, 0)$ is not valid in 2D homogeneous coordinates.

This also works in 3D coordinates where (X, Y, Z, W) in homogeneous coordinates represents the point $(X/W, Y/W, Z/W)$ in Cartesian coordinates.

Homogeneous Coordinates

Note that the homogeneous coordinates for points are not unique since $\boldsymbol{x}$ and $\alpha\boldsymbol{x}$ represent the same point if $\alpha \neq 0$. This creates an equivalence class defined by relation $\sim$:

$$\boldsymbol{p} \sim \boldsymbol{q} \iff (\exists \alpha \neq 0)(\alpha \boldsymbol{p} = \boldsymbol{q}). \tag{1}$$

Note that this sort of equivalence will also hold for most of the matrices we will work with as for $\boldsymbol{y} = A\boldsymbol{x}$ $\alpha\boldsymbol{y} = \alpha(A\boldsymbol{x}) = (\alpha A)\boldsymbol{x}$.

Basic Pinhole Camera Model

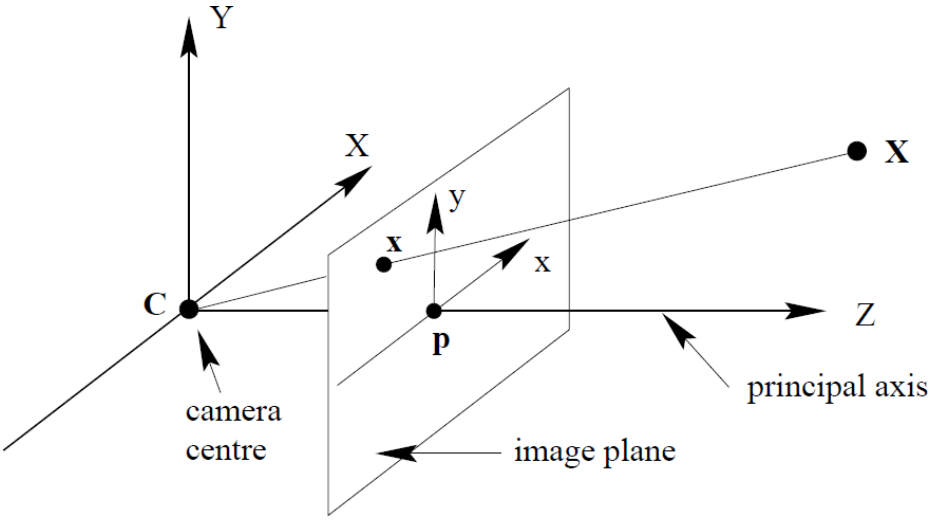


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Simple Pinhole Camera Model

Using homogeneous coordinates we can express the simple pinhole camera model using the following equation:

$$P\boldsymbol{X} = K\Pi_0\boldsymbol{X} = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2}$$

where f is the **focal length** (x_0, y_0) is the **principal point**.

Camera Rotation and Translation

So far we have consider a coordinate system where the camera center is in the origin, the optical axis is the z axis and the remaining two axes are aligned with the image coordinates. However, we will often need to consider camera which has its own rotation R and translation $\boldsymbol{t}$ w.r.t the world coordinate system. In such case we may rewrite the equation into the form:

$$P\boldsymbol{X} = K[R|\boldsymbol{t}]\boldsymbol{X}. \tag{3}$$

We then call the matrix K the **intrinsic matrix** or also the calibration matrix or the camera matrix and $[R|\boldsymbol{t}]$ the **extrinsic matrix**. Note that $\boldsymbol{t}$ is not the same as camera center, which can be calculated as $\boldsymbol{C} = -R^T\boldsymbol{t}$.

Intrinsic Camera Matrix

In a more general case K can be slightly more complicated:

$$K = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

where s is skew of the lens and f_x and f_y are different focal lengths for their corresponding directions. However, for most modern cameras we can assume that $f_x = f_y$, $s = 0$ and (x_0, y_0) lies in the center of the image.

Radial Distortion

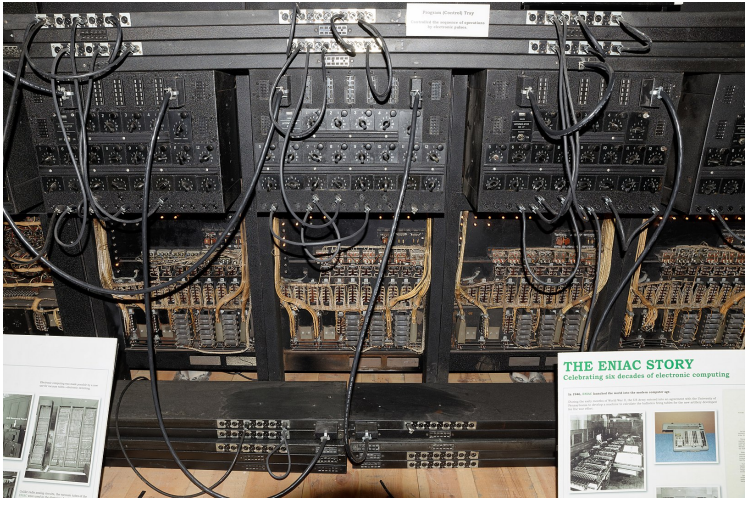


Image adopted from: Jud McCranie. *Wikimedia Commons*.

Radial Distortion

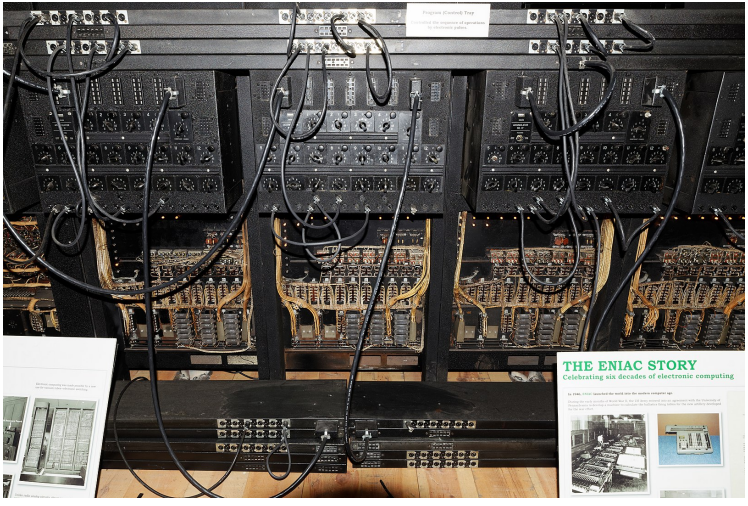


Image adopted from: Jud McCranie. *Wikimedia Commons*.

Radial Distortion

To correct for the distortion we can calculate the undistorted position of the pixels $(\hat{x}, \hat{y})$:

$$\hat{x} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \tag{5}$$

$$\hat{y} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \tag{6}$$

where k_1, k_2, k_3 are the parameters of the model and $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$. In some cases it is sufficient to assume that k_3 or even k_2 is zero. The mappings (5) and (6) can be used to transform the captured image to remove the effects of the distortion. Radial distortion is usually most pronounced in cheaper cameras or cameras with special lens.

Triangulation

Let us consider a point seen in two images as $\boldsymbol{x}$ and $\boldsymbol{x'}$ with projection matrices P and P'. Our task is to find $\boldsymbol{X}$. We can achieve this using triangulation. To find it we will construct use a linear system in $\boldsymbol{X}$ from equations:

$$\boldsymbol{x} \sim P\boldsymbol{X} \quad \boldsymbol{x'} \sim P'\boldsymbol{X}. \tag{7}$$

The problem is that in reality, due to noise and other factors the system often won't have an exact solution.

Triangulation

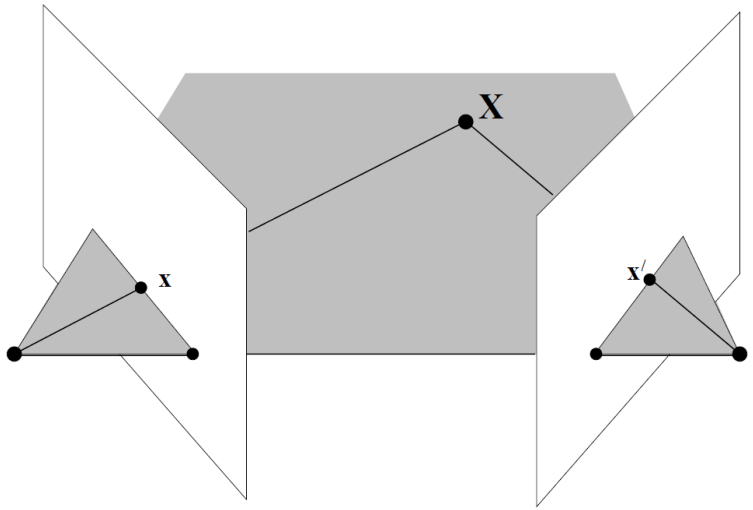


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Solution

We can obtain the linear system by considering equations $\boldsymbol{x} \times P\boldsymbol{X} = \boldsymbol{0}$ to form:

$$A\boldsymbol{X} = \begin{pmatrix} x\boldsymbol{p}^T_{(3,:)} - \boldsymbol{p}^T_{(1,:)} \\ y\boldsymbol{p}^T_{(3,:)} - \boldsymbol{p}^T_{(2,:)} \\ x'\boldsymbol{p}'^T_{(3,:)} - \boldsymbol{p}'^T_{(1,:)} \\ y'\boldsymbol{p}'^T_{(3,:)} - \boldsymbol{p}'^T_{(2,:)} \end{pmatrix} \boldsymbol{X} = \boldsymbol{0}, \tag{8}$$

where $\boldsymbol{x} = (x, y, 1)^T, \boldsymbol{x'} = (x', y', 1)^T, \boldsymbol{p}_{(i,:)}$ is the i-th row of P and analogously for P'. Since the problem may be overdetermined we find $\boldsymbol{X}$ as the singular vector corresponding to the smallest singular value using SVD.

Optimizing for Geometric Error

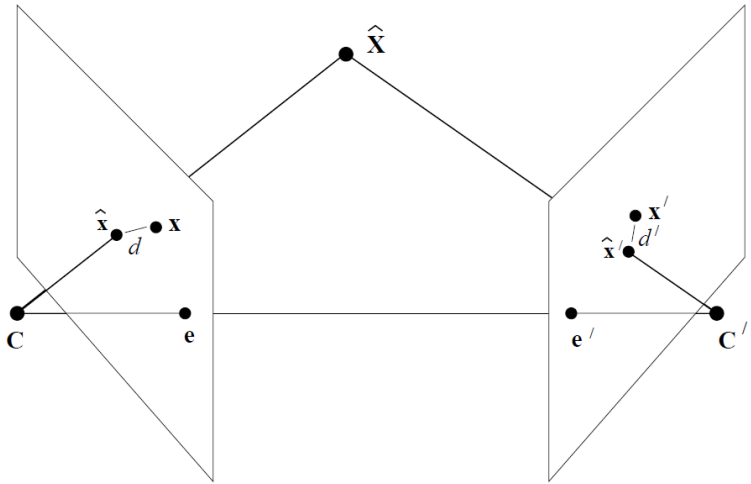


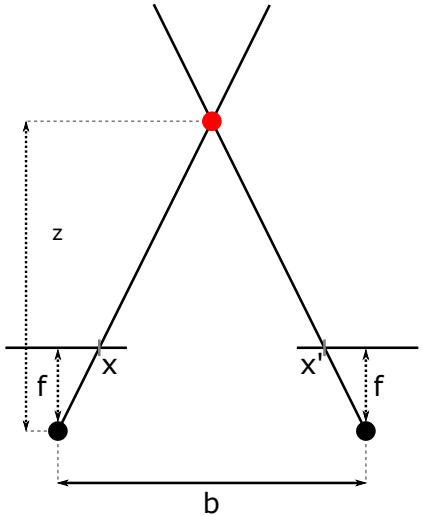
Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Depth in Aligned Stereo

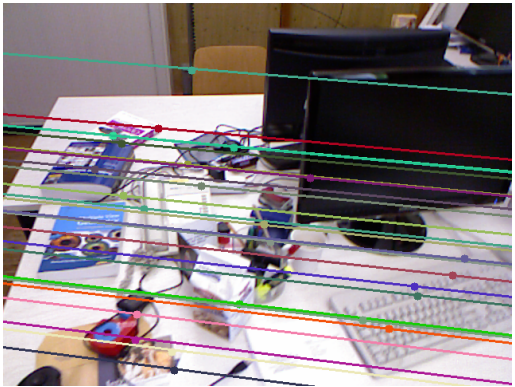
On previous slides we dealt with the general triangulation, but if we a two cameras with the same orientation and horizontal alignment we can calculate the depth of a correspondence $(x, y) - (x', y)$ using the formula:

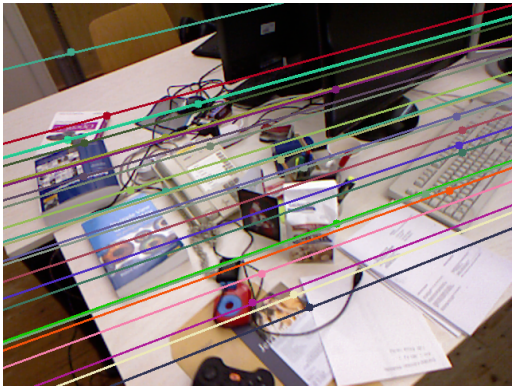
$$Z(x, x') = \frac{bf}{|x - x'|}, \qquad (9)$$

where f is the focal length, b is the baseline distance.

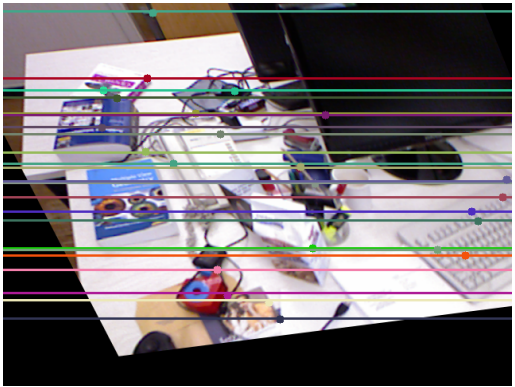


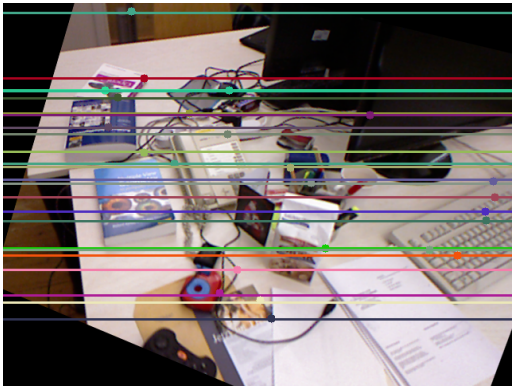
Stereo Rectification





Stereo Rectification





Disparity map

After rectifying both images. We can calculate the depth by calculating the disparity map. Recall that we defined it as map $d(x, y)$ such that in row y we try to find

$$x + d(x, y) = x', \tag{10}$$

such that $(x, y)^T$ and $(x', y)^T$ are correspondences. If we succeed we can use

$$Z(x, y) = \frac{bf}{|x - x'|} = \frac{bf}{|d(x, y)|}. \tag{11}$$

Then we can obtain the 3D point $\boldsymbol{X} = (X, Y, Z)$ as:

$$\boldsymbol{X} = ZK^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{12}$$

Finding d using block matching

We can compare how similar different blocks from one image are in another one using normalized cross-correlation. We can use this to compute the disparity:

$$d(x, y) = \arg\max_{d} \mathrm{NCC}(i, i', x, y, d, n). \tag{13}$$

Note that n determines how large is the neighborhood around points we consider for matching. We have to select n carefully. If we use n too small we can get more detail, but also more noise. With larger n we get smoother disparity maps, but less detail.