Autonomous Vision Group
Prof. Dr.-Ing. Andreas Geiger

Eberhard Karls
Universität
Tübingen

Self-Driving Cars

Ex. 6 – Pen and Paper (Lecture 07 - 12)

6.1 3D Geometry

a) Given a 3D point $\mathbf{x}_w \in \mathbb{R}^3$, such that $\mathbf{x}_w = \begin{bmatrix} 3 \\ 2 \\ 6 \end{bmatrix}$ write $\mathbf{x}_w$ as an augmented vector $\bar{\mathbf{x}}_w$ and as a homogeneous vector $\tilde{\mathbf{x}}_w$.

$$\bar{\mathbf{x}}_w = \begin{bmatrix} 3 \\ 2 \\ 6 \\ 1 \end{bmatrix} \text{ and } \tilde{\mathbf{x}}_w = s \cdot \begin{bmatrix} 3 \\ 2 \\ 6 \\ 1 \end{bmatrix} \text{ where, } s \in \mathbb{R}.$$

b) Consider a camera C with pose defined by the rotation and translation matrices,

$$\mathbf{R} = \begin{bmatrix} 0.38 & -0.82 & 0.42 \\ 0.87 & 0.17 & -0.45 \\ 0.29 & 0.54 & 0.78 \end{bmatrix} \text{ and } \mathbf{t} = \begin{bmatrix} 1.3 \\ 2.0 \\ -1.5 \end{bmatrix}.$$

The camera C has focal lengths $(f_x, f_y) = (785, 786)$, skew $s = 0$ and camera center $(c_x, c_y) = (680, 680)$. Project the point $\mathbf{x}_w$ to the image plane of camera C.

The intrinsic matrix of camera C is,

$$\mathbf{K} = \begin{bmatrix} 785 & 0 & 680 \\ 0 & 786 & 680 \\ 0 & 0 & 1 \end{bmatrix}$$

and the projection onto the image plane of camera C,

$$\tilde{\mathbf{x}}_s = \begin{bmatrix} \mathbf{K} & \mathbf{0} \end{bmatrix} \bar{\mathbf{x}}_c = \begin{bmatrix} \mathbf{K} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & \mathbf{1} \end{bmatrix} \bar{\mathbf{x}}_w = \begin{bmatrix} 6094.6 \\ 5256.0 \\ 5.13 \end{bmatrix}$$

$$\bar{\mathbf{x}}_s = \begin{bmatrix} 1188.03 \\ 1024.73 \\ 1.0 \end{bmatrix}$$

6.2 Epipolar Geometry and Bundle Adjustment

We are given a stereo camera setup where the first camera is on the left of the second camera and the relative rotation and translation matrices are given by,

$$\mathbf{R} = \mathbf{I} \text{ and } \mathbf{t} = \begin{bmatrix} -0.5 \\ 0 \\ 0 \end{bmatrix}.$$

a) Given the point $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ on the image plane of the left camera, compute the (full) set of points on the image plane of the right camera which fulfil the epipolar constraint?

The essential matrix $\tilde{\mathbf{E}}$ is given by,

$$\tilde{\mathbf{E}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.5 \\ 0 & -0.5 & 0 \end{bmatrix}$$

Thus, for a point $\mathbf{x}_1$ on the left image and the point $\mathbf{x}_2$ on the right image which fulfil the epipolar constraint,

$$\begin{aligned} \bar{\mathbf{x}}^{2\,\mathrm{T}} \tilde{\mathbf{E}} \bar{\mathbf{x}}_1 &= 0 \\ \Rightarrow \begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.5 \\ 0 & -0.5 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} &= 0 \\ \Rightarrow 0.5y' - 0.5y &= 0 \\ \Rightarrow y' &= y \end{aligned}$$

Thus, given the point $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, the set of points given by $y' = 2$ fulfil the epipolar constraint.

b) Does the point $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ on the image plane of the left camera and the point $\mathbf{x}_2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$ on the image plane of the right camera fulfil the epipolar constraint?

No, as $\mathbf{x}_2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$ does not lie on $y' = 2$.

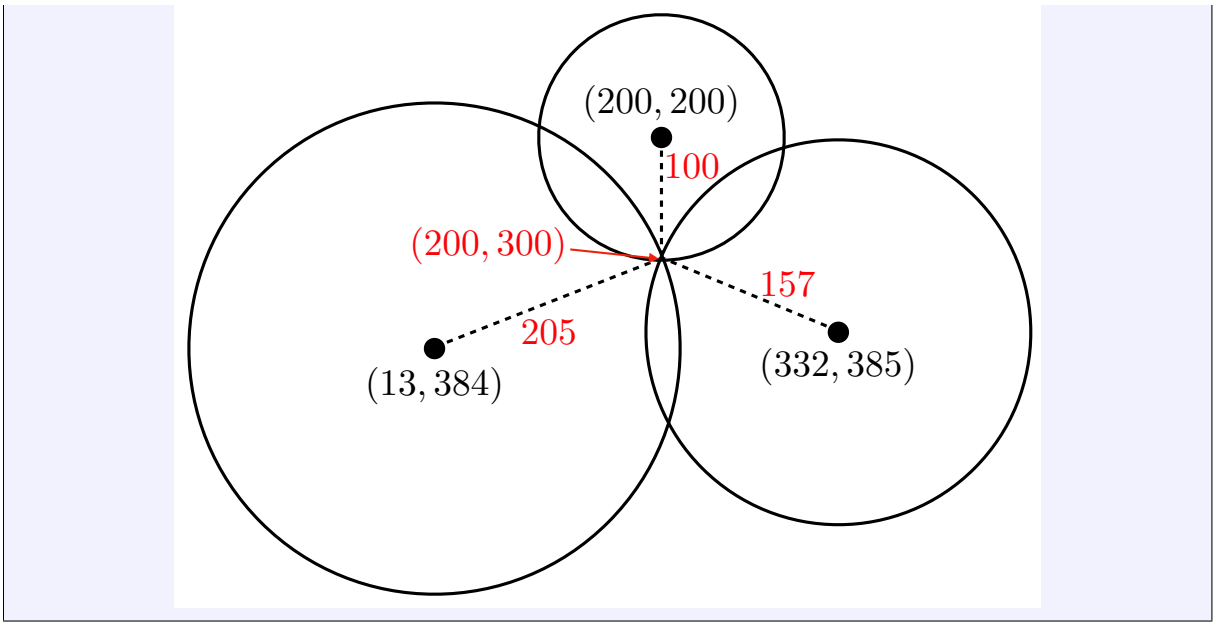
c) Given N cameras with $\prod = \{\boldsymbol{\pi}_i\}$ their intrinsic and extrinsic parameters, $\mathcal{X}_w = \{\mathbf{x}_p^w\}$ with $\mathbf{x}_p^w \in \mathbb{R}^3$ denote a set of P 3D points in world coordinates and $\mathcal{X}_s = \{\mathbf{x}_{ip}^s\}$ with $\mathbf{x}_{ip}^s \in \mathbb{R}^2$ denote the image (screen) observations in all i cameras. Give an example of a scenario where the bundle adjustment error is 0?

When all points are noise-free projected into the images, and the intrinsic and extrinsic parameters $\mathcal{X}_w = \{\mathbf{x}_p^w\}$ are the ground truth.

6.3 Localization

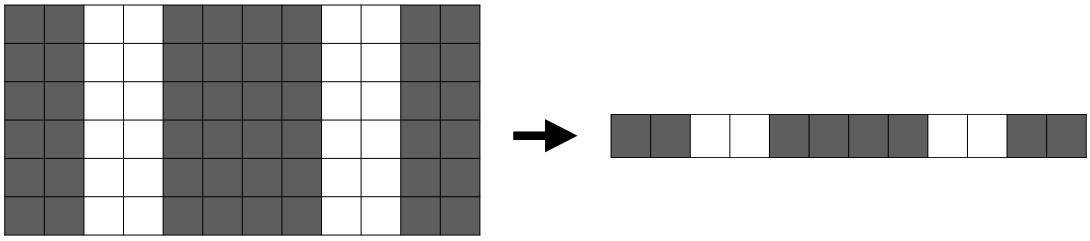
Imagine that you reside in the 2-dimensional flatworld. The residents of flatworld use FPS (Flatworld Positioning System) with ground based beacons for localization. One day while on a hike through the forests of flatworld you take a wrong turn and thus need to use the services of FPS to find your way back home. You find that the nearest FPS beacons (alpha, beta, gamma) are located at grid positions $(200, 200)$, $(332, 385)$ and $(13, 384)$. Your hand-held FPS measures a time delay of 3.345×10^{-4} seconds from beacon alpha, a time delay of 5.251×10^{-4} seconds from beacon beta and a time delay of 6.857×10^{-4} seconds from beacon gamma to receive FPS signals transmitted by beacons alpha, beta and gamma respectively. Can you use this time delay information to find your position in flatworld? Note, flatworld grid positions are defined in kilometers and the speed of light in flatworld is 2.99×10^8 m/s. For computing your position in flatworld, round distance measurements to the nearest kilometer.

The beacons alpha, beta and gamma are located at distances of 100, 157 and 205 kilometers respectively. Solving for the point of intersection of the circles of 100, 157 and 205 kilometer radii with centres at $(200, 200)$, $(332, 385)$ and $(13, 384)$ yields the location at $(200, 300)$.



6.4 Lane Detection

Consider the simple (6×12) binary image of a road given below,



a) First, apply a simple gradient filter (e.g., the one discussed in the lecture) to detect the left and right lanes. For simplicity, apply the gradient filter to the 1D image row shown on the right and calculate the resultant image. Use zero padding if necessary.

We use the simple gradient filter,

-1	0	1

Using zero padding of 1 at borders of the 1D lane image and applying the gradient filter,

0	1	1	-1	-1	0	0	1	1	-1	-1	0

b) Next, we will explore end to end lane detection with a clustering loss. The clustering loss is defined as,

$$L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}\|\mu_c - x_i\|_2^2$$

$$L_{dist} = \frac{1}{C(C-1)}\sum_{c_A=1}^{C}\sum_{c_B=1, c_A\neq c_B}^{C}\|\mu_{c_A} - \mu_{c_B}\|_2^2$$

where, C is the number of classes, N_c are the number of points (pixels) in class c, μ_c is the mean feature embedding for class c and $\mathbf{x}_i$ is the feature embedding of a point (pixel) in class c. Given the following features for the 1D image row (from a feature extractor),

0	0	0.8	1.1	0	0	0	0	1.9	2.3	0	0

compute the clustering loss $\frac{L_{var}}{L_{dist}}$ for the following two cases with $C = 3$ (known apriori), $\mu_1 = 0, \mu_2 = 0.2, \mu_3 = 1.5$ and $\mu_1 = 0, \mu_2 = 1.0, \mu_3 = 2.0$. Assign each point (pixel) in the image row to the nearest cluster center.

For the first case with $\mu_1 = 0, \mu_2 = 0.2, \mu_3 = 1.5$, we have,

$$L_{var} = \frac{1}{3}\Big(\frac{1}{8}(0-0)^2 + \frac{1}{1}(0.8-0.2)^2 + \frac{1}{3}\big((1.1-1.5)^2 + (1.9-1.5)^2 + (2.3-1.5)^2\big)\Big) = 0.227$$

$$L_{dist} = \frac{1}{3\times 2}\Big(2(0-0.2)^2 + 2(0-1.5)^2 + 2(0.2-1.5)^2\Big) = 1.327$$

$$\frac{L_{var}}{L_{dist}} = 0.171$$

For the second case with $\mu_1 = 0, \mu_2 = 1.0, \mu_3 = 2.0$, we have,

$$L_{var} = \frac{1}{3}\Big(\frac{1}{8}(0-0)^2 + \frac{1}{2}\big((0.8-1.0)^2 + (1.1-1.0)^2\big) + \frac{1}{2}\big((1.9-2.0)^2 + (2.3-2.0)^2\big)\Big) = 0.025$$

$$L_{dist} = \frac{1}{3\times 2}\Big(2(0-1.0)^2 + 2(0-2.0)^2 + 2(1.0-2.0)^2\Big) = 2.0$$

$$\frac{L_{var}}{L_{dist}} = 0.0125$$

6.5 Stereo Matching

Consider a stereo camera setup where we want to estimate the disparity, where there is a change in brightness between the left and right camera due to different exposure settings. A row of pixels from the left camera and the corresponding row from the right camera are given below,

19	0	12	2	13	22	20	26

Left Camera

11	19	18	23	18	18	25	24

Right Camera

Consider the pixel highlighted in red in the left camera, whose true disparity is 4. We would like to estimate the disparity of this pixel using block matching with a window size of 3 (1×3 as we consider a 1D pixel row).

a) Is the true disparity recovered if we use the sum of squared differences (SSD) similarity metric?

SSD leads to a incorrect disparity of 1. SSD values given disparity,

Disparity	0	1	2	3	4	5
SSD	45	21	69	89	22	186

Thus, the disparity value with the minimum SSD is 1.

We also show the steps involved in calculating the SSD for the disparity value of $d = 4$ (the steps involved in computing the SSD for other disparity values follows analogously). The SSD at position x on a 1D pixel row, for a disparity d is given by,

$$\text{SSD}(x, d) = \|\mathbf{w}_L(x) - \mathbf{w}_R(x-d)\|_2^2,$$

where, $\mathbf{w}_L(x), \mathbf{w}_R(x-d)$ are 1×3 windows centred at x and $x-d$ respectively, represented as

vectors. We want to compute the disparity at position $x = 6$,

$$\mathbf{w}_L(6) = \begin{bmatrix} 22 \\ 20 \\ 26 \end{bmatrix}, \quad \mathbf{w}_R(2) = \begin{bmatrix} 19 \\ 18 \\ 23 \end{bmatrix}, \quad \text{SSD}(6,2) = 22.$$

b) Which similarity metric would you use instead considering the brightness changes between the left and right images? Show that the proposed metric can recover the true disparity for the pixel highlighted in red.

ZNCC correctly recovers the true disparity. ZNCC values given disparity,

Disparity	0	1	2	3	4	5
ZNCC	0.05	0.94	-0.18	-0.76	0.99	0.07

Thus, the disparity value with the maximum ZNCC is 4.

We also show the steps involved in calculating the ZNCC for the disparity value of $d = 4$ (the steps involved in computing the ZNCC for other disparity values follows analogously). The ZNCC at position x on a 1D pixel row, for a disparity d is given by,

$$\text{ZNCC}(x,d) = \frac{\left(\mathbf{w}_L(x) - \bar{\mathbf{w}}_L(x)\right)^T \left(\mathbf{w}_R(x-d) - \bar{\mathbf{w}}_R(x-d)\right)}{\|\mathbf{w}_L(x) - \bar{\mathbf{w}}_L(x)\|_2 \, \|\mathbf{w}_R(x-d) - \bar{\mathbf{w}}_R(x-d)\|_2}$$

where, $\mathbf{w}_L(x), \mathbf{w}_R(x-d)$ are 1×3 windows centred at x and $x-d$ respectively, represented as vectors, and $\bar{\mathbf{w}}_L(x), \bar{\mathbf{w}}_R(x-d)$ are the corresponding means of the windows multiplied by a vector of ones ($\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top$). The disparity at position $x = 6$,

$$\mathbf{w}_L(6) = \begin{bmatrix} 22 \\ 20 \\ 26 \end{bmatrix}, \quad \mathbf{w}_R(2) = \begin{bmatrix} 19 \\ 18 \\ 23 \end{bmatrix}, \quad \bar{\mathbf{w}}_L(6) = \begin{bmatrix} 22.66 \\ 22.66 \\ 22.66 \end{bmatrix}, \quad \bar{\mathbf{w}}_R(2) = \begin{bmatrix} 20.00 \\ 20.00 \\ 20.00 \end{bmatrix}, \quad \text{ZNCC}(6,2) = 0.99.$$

6.6 Depth Estimation Error

Given the focal length f and baseline b for the left camera of a stereo camera setup and the disparity d, the depth can be calculated using the simple formula, $z = \frac{fb}{d}$. Show that for an error of ϵ_d in the estimated disparity, the corresponding error in the obtained depth ϵ_z is quadratic in depth,

$$\epsilon_z = \frac{z^2}{bf}\epsilon_d$$

The depth error can be written as,

$$\begin{aligned} \epsilon_z &= \frac{bf}{d} - \frac{bf}{d + \epsilon_d} \\ &= \frac{z^2 \epsilon_d}{bf + z\epsilon_d} \end{aligned}$$

Using the first order Taylor series approximation of this error around $\epsilon_d = 0$ (generally the disparity error is small),

$$\epsilon_z(\epsilon_d) = \frac{z^2 \epsilon_d}{bf + z\epsilon_d}$$

and,

$$\epsilon_z(\epsilon_d) \approx_{\epsilon_d = 0} \epsilon_z(0) + \frac{d\epsilon_z(0)}{d\epsilon_d}\epsilon_d$$

where,

$$\frac{d\epsilon_z}{d\epsilon_d} = \frac{bfz^2}{(bf + z\epsilon_d)^2}$$

thus,

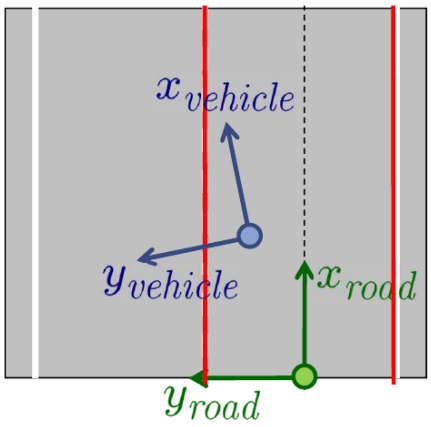
$$\frac{d\epsilon_z}{d\epsilon_d}\,|_{\epsilon_d=0} = \frac{z^2}{bf}$$

So, the error in depth ϵ_z when the error in disparity ϵ_d is small,

$$\epsilon_z = \frac{z^2\epsilon_d}{bf}$$

6.7 Model for Straight Lanes

Consider the coordinate system defined for straight roads and vehicles in the lecture (also shown on the left), where the road coordinate system is centred at the road and the vehicle coordinate system is centred at the vehicle. The road is of width B and thus the left/right lane markings at positions $\pm\frac{B}{2}$ from the center of the lane. The lateral and longitudinal displacements of the vehicle are denoted by d_{lat}, d_{long} and the yaw angle by ψ. For a path length of l, a point on the left or right lanes $\begin{bmatrix} l \\ \pm\frac{B}{2} \end{bmatrix}$ in the road coordinate system can be transformed to the vehicle coordinate system as,



$$\begin{bmatrix} x_{vehcile} \\ y_{vehcile} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} l - d_{long} \\ \pm\frac{B}{2} - d_{lat} \end{bmatrix}$$

Show that for $\psi \approx 0$, this transformation can be simplified to,

$$y_{vehcile} = -\psi\, x_{vehicle} \pm \frac{B}{2} - d_{lat}.$$

The transformation for a point on left lane $\begin{bmatrix} l \\ \frac{B}{2} \end{bmatrix}$ can be simplified as follows,

$$\begin{bmatrix} x_{vehcile} \\ y_{vehcile} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} l - d_{long} \\ \frac{B}{2} - d_{lat} \end{bmatrix}$$

gives us two equations,

$$(1) \Rightarrow x_{vehicle} = \cos(\psi)(l - d_{long}) + \sin(\psi)(\frac{B}{2} - d_{lat})$$
$$(2) \Rightarrow x_{vehicle} = -\sin(\psi)(l - d_{long}) + \cos(\psi)(\frac{B}{2} - d_{lat})$$

Using (1) to eliminate l,

$$l = \frac{x_{vehicle} - \sin(\psi)(\frac{B}{2} - d_{lat}) + \cos(\psi)d_{long}}{\cos(\psi)}$$

Plugging the l into (2),

$$y_{vehicle} = -\sin(\psi)\Big(\frac{x_{vehicle} - \sin(\psi)(\frac{B}{2} - d_{lat})}{\cos(\psi)}\Big) + \cos(\psi)(\frac{B}{2} - d_{lat})$$
$$= -\tan(\psi)(x_{vehicle} - \sin(\psi)\frac{B}{2} + \sin(\psi)d_{lat}) + \cos(\psi)(\frac{B}{2} - d_{lat})$$

If $\psi \approx 0$, then $\sin(\psi) \approx 0$, $\cos(\psi) \approx 1$ and $\tan(\psi) \approx \psi$. Thus,

$$y_{vehicle} = -\psi\, x_{vehicle} + \frac{B}{2} - d_{lat}.$$

Similarly for a point $\begin{bmatrix} l \\ -\frac{B}{2} \end{bmatrix}$ on the right lane we can show that,

$$y_{vehicle} = -\psi\, x_{vehicle} + \frac{B}{2} - d_{lat}.$$

6.8 Clothoids

As introduced in the lecture, a clothoid is a curve whose curvature κ linearly depends upon arc length l,

$$\frac{1}{r(l)} = \kappa(l) = \kappa_0 + \kappa_1 l$$

Using the coordinate system specified in the lecture,

a) Can you represent a straight line using a clothoid? Provide an example of a straight line and the appropriate values for κ_0, κ_1.

To represent the straight line $y = 0$, set $\kappa_0 = \kappa_1 = 0$. A point with arc length l on the clothoid will be at position $l, 0$ on the straight line $y = 0$.

b) Can you represent a circle using a clothoid? Provide an example of a circle and appropriate values for κ_0, κ_1.

To represent the circle $x^2 + y^2 = r^2$, set $\kappa_0 = \frac{1}{r}, \kappa_1 = 0$. The yaw angle with respect to the origin for a point with arc length l on the clothoid,

$$\psi(l) = \int_0^l \kappa_0 d\lambda = \frac{l}{r}.$$

Thus, the position of a point with arc length l on the clothoid,

$$x(l) = \int_0^l \cos\left(\frac{\lambda}{r}\right) d\lambda = r \sin\left(\frac{l}{r}\right).$$

$$y(l) = \int_0^l \sin\left(\frac{\lambda}{r}\right) d\lambda = r - r\cos\left(\frac{l}{r}\right).$$

6.9 Linearization

In order to solve non-linear optimization problems such as the Horn-Schunck formulation of optical flow, a popular approach is to use linearization. To better understand this approach, we will consider two simple 1D problems.

a) First, we consider the simple (1D) non-linear minimization problem,

$$\underset{x}{\text{argmin}}\ g(x) = \underset{x}{\text{argmin}}\ (x^2 - 5)^2$$

To find the optimum value of x follow the minimization technique introduced in the lecture, i.e., iteratively linearize the function x^2 and minimize the resultant simpler quadratic function. Use $x = 3$ as the initial value and 3 iterations should be sufficient to find a good solution. (For a more general overview of Gauss-Newton methods refer to the article here).

Starting with $x = 3$, we linearlize $h(x) = x^2$ (such that, $g(x) = (h(x) - 5)^2$),

$$h(x) = x^2 \overset{3}{\approx} 9 + 6(x - 3)$$

Plugging this into $g(x)$,

$$\underset{x}{\text{argmin}}\ h_1'(x) = \underset{x}{\text{argmin}}\ (9 + 6(x - 3) - 5)^2 = \frac{7}{3}$$

Repeating the steps above with $x = \frac{7}{3}$,

$$\underset{x}{\text{argmin}}\ h_2'(x) = \underset{x}{\text{argmin}}\ \left(\left(\frac{7}{3}\right)^2 + \frac{14}{3}\left(x - \frac{7}{3}\right) - 5\right)^2 = 2.238$$

Repeating the steps above with $x = 2.238$,

$$\underset{x}{\text{argmin}}\ h_3'(x) = \underset{x}{\text{argmin}}\ (2.238^2 + 4.476(x - 2.238) - 5)^2 = 2.236$$

This result is close to the optimum which is $x = \sqrt{5} \approx 2.236$.

b) Next, we consider the following simple (1D) image warping problem, where the image I is warped by the function $u(x)$ to match the image I' (here x denotes a position in the image),

$$L_{warp} = \int \left(I(x + a\,u(x)) - I'(x)\right)^2 dx$$

where a is a constant. Propose a linearization of the objective L_{warp} which has a unique optimum (of $u(x)$) when I, I' are discrete. Under which conditions is the proposed linearization valid?

We can approximate $I(x + a\,u(x))$ using a Taylor series approximation when a and $u(x)$ are small. The (linear) Taylor series approximation is given by (ignoring higher order terms),

$$I(x + a\,u(x)) \overset{x}{\approx} I(x) + I_x(x)(a\ u(x))$$

where I_x is the derivative of I with respect to x. Plugging this back into L_{warp},

$$L_{warp} = \int \left(I(x) + a\ u(x) I_x(x) - I'(x)\right)^2 dx$$

If I, I' are discrete we can rewrite the objective as,

$$\bar{L}_{warp} = \sum_{i=1}^{N} \left(a\ u(x_i) I_x(x_i) + I(x_i) - I'(x_i)\right)^2$$

where the 1D signals I, I' have length N. As $\bar{L}_{warp}$ is of the sum of squares form, it has a unique optimum.

6.10 Kalman Filtering

We will now use a Kalman filter to estimate the state $\mathbf{x}_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix}$, i.e. the position and velocity, of a moving ball. We will assume simple 1D linear motion as shown below,

where $y_t = \begin{bmatrix} p_t \end{bmatrix}$ are the observations at each timestep t. As introduced in the lecture, Kalman filtering consists of two steps which are described using the following Kalman filter equations. The first step is the **prediction step** where we predict the mean $\bar{x}_t$ and covariance $\bar{P}_t$ of the state one timestep ahead, given x_{t-1} and P_{t-1},

$$\hat{\mathbf{x}}_t = \mathbf{F}\mathbf{x}_{t-1}$$
$$\hat{\mathbf{P}}_t = \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^{\mathrm{T}}$$

where $\mathbf{F}$ is the state transition matrix and $\mathbf{H}$ is the observation matrix. The second step is the **correction step** which corrects our estimate of the state x_t using the observations y_t, using the following equations,

$$\mathbf{K}_t = \hat{\mathbf{P}}_t\mathbf{H}^{\mathrm{T}}(\mathbf{H}\hat{\mathbf{P}}_t\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$$
$$\mathbf{x}_t = \hat{\mathbf{x}}_t + \mathbf{K}_t(\mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_t)$$
$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\hat{\mathbf{P}}_t$$

where $\mathbf{K}_t$ is the Kalman gain and R is the sensor/observation noise.

a) Determine $\mathbf{F}$ and $\mathbf{H}$ if we assume a simple constant velocity model for the motion of the ball.

$$\mathbf{F} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

b) Use the Kalman filtering equations described above to determine $\mathbf{x}_t, \mathbf{P}_t$ for $t = \{2, 3\}$. Assume that the initial state $\mathbf{x}_t = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$ is given and $\mathbf{R} \approx 0$.

The states $\mathbf{x}_2, \mathbf{P}_2$ and $\mathbf{x}_3, \mathbf{P}_3$ can be obtained through the straight-forward application of the Kalman filtering equations described above. In detail, for $t = 2$,

$$\hat{\mathbf{x}}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \text{ and } \hat{\mathbf{P}}_2 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

$$\mathbf{K}_2 = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}, \ \mathbf{x}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \text{ and } \mathbf{P}_2 = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

For $t = 3$,

$$\hat{\mathbf{x}}_3 = \begin{bmatrix} 4 \\ 2 \end{bmatrix} \text{ and } \hat{\mathbf{P}}_3 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{K}_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ \mathbf{x}_3 = \begin{bmatrix} 5 \\ 3 \end{bmatrix} \text{ and } \mathbf{P}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

6.11 Greedy Matching

We are given the following matching scores for 5 detections from frame i to $i+1$. Use the greedy matching algorithm introduced in the lecture to match detections across frames i to $i+1$.

i \ $i+1$	1	2	3	4	5
1	0.98	0.42	0.11	0.06	0.29
2	0.18	0.30	0.04	0.90	0.51
3	0.95	0.45	0.31	0.15	0.61
4	0.09	0.80	0.15	0.17	0.85
5	0.77	0.55	0.13	0.49	0.69

What is the value of the greedy solution?

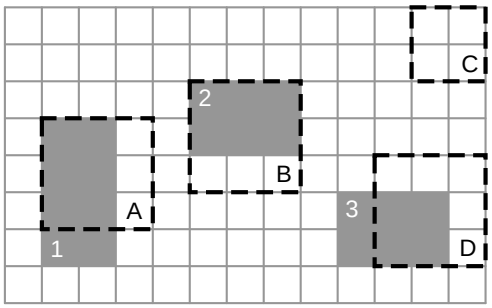
Greedy solution is,

i \ $i+1$	1	2	3	4	5
1	**0.98**	~~0.42~~	~~0.11~~	~~0.06~~	~~0.29~~
2	~~0.18~~	~~0.30~~	~~0.04~~	**0.90**	~~0.51~~
3	~~0.95~~	~~0.45~~	~~0.31~~	~~0.15~~	**0.61**
4	~~0.09~~	**0.80**	~~0.15~~	~~0.17~~	~~0.85~~
5	~~0.77~~	~~0.55~~	**0.13**	~~0.49~~	~~0.69~~

The greedy solution has a total score of 3.42.

6.12 Detection Performance

Consider the following "image" where the three gray shaded boxes (1,2,3) correspond to the true objects and the four dashed boxes (A,B,C,D) correspond to detections of an object detector:



a) Calculate the IoU (Intersection-over-Union) metric for the pairs $(A,1)$, $(B,2)$ and $(D,3)$. Assuming a detection threshold of $\tau = 0.5$, which of the three objects has been correctly detected?

IoU is given by,

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}.$$

$\text{IoU}(A,1) = \frac{6}{11} = 0.55$, $\text{IoU}(B,2) = \frac{6}{9} = 0.67$, $\text{IoU}(D,3) = \frac{4}{11} = 0.36$. Objects A and B have been correctly detected.

b) Calculate the number of true positives (TP), false positives (FP) and false negatives (FN) assuming the same detection threshold as in the previous question ($\tau = 0.5$).

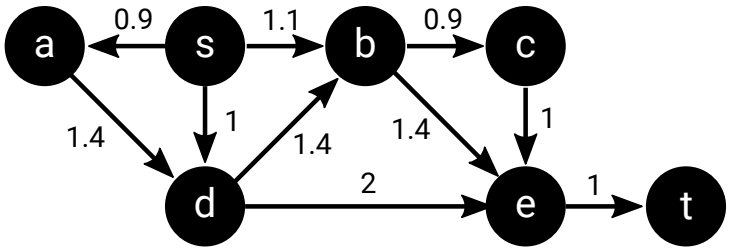
TP $\Rightarrow$ 2, FP $\Rightarrow$ 2, FN $\Rightarrow$ 1.

c) Calculate precision and recall at a detection threshold of $\tau = 0.6$.

TP $\Rightarrow$ 1, FP $\Rightarrow$ 3, FN $\Rightarrow$ 2. Precision is $\frac{1}{1+3} = 0.25$ and recall is $\frac{1}{1+2} = 0.33$.

6.13 Dijkstra's Shortest Path Algorithm

Find the shortest path from source to sink in the following weighted graph using Dijkstra's algorithm. Specify the elements in the *open min heap* and the *closed set* for each step of the algorithm. Upon completion, illustrate the shortest path by marking the corresponding edges in the graph.

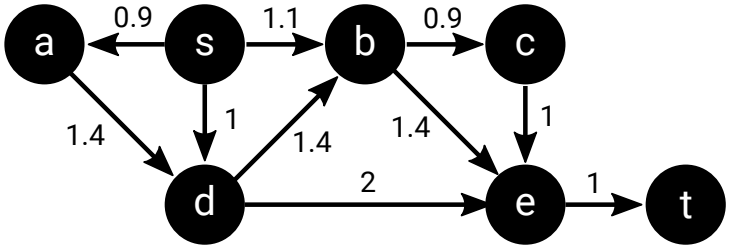


Step 1	Open Min Heap: s Cost: 0	Closed Set: –
Step 2	Open Min Heap: a, d, b Cost: 0.9, 1, 1.1	Closed Set: s
Step 3	Open Min Heap: d, b Cost: 1, 1.1	Closed Set: s, a
Step 4	Open Min Heap: b, e Cost: 1.1, 3	Closed Set: s,a,d
Step 5	Open Min Heap: c, e Cost: 2.0, 2.5	Closed Set: s,a,d,b
Step 6	Open Min Heap: e Cost: 2.5	Closed Set: s,a,d,b,c
Step 7	Open Min Heap: t Cost: 3.5	Closed Set: s,a,d,b,c,e

Final shortest path is, s → b → e → t.

6.14 A* Algorithm

Find the shortest path from source to sink in the following weighted graph using the A* algorithm and a Euclidean distance as planning heuristics (all numbers below denote Euclidean distances). Specify the elements in the *open min heap* and the *closed set* for each step of the algorithm. Upon completion, illustrate the shortest path by marking the corresponding edges in the graph.



Distances:
a -> t = 4.1
s -> t = 3.2
b -> t = 2.2
c -> t = 1.4

Step 1	Open Min Heap: s Cost: 3.2	Closed Set: –
Step 2	Open Min Heap: b,d,a Cost: 3.3,4,5	Closed Set: s

Step 3

Open Min Heap	c,e,d,a
Cost	3.4,3.5,4,5

Closed Set	s,b

Step 4

Open Min Heap	e,d,a
Cost	3.5,4,5

Closed Set	s,b,c

Step 4

Open Min Heap	t,d,a
Cost	3.5,4,5

Closed Set	s,b,c,e

Final shortest path is, s → b → e → t.