Problem Set 1 Solutions

1. Line detection

1 The range of slope and y-intercept are infinity.

2 Line appears as sinusoid curve. $d = x\cos\theta + y\sin\theta$

3 $x\cos(\theta) + y\sin(\theta) - d = 0 \Rightarrow d = 1\cos(0) + 1\sin(0) = 1$. Therefore, it casts a vote at location $(0, 1)$ in Hough space.

4 Each edge point is transformed to a line in the Hough space, and the areas where most lines intersect in the Hough space is interpreted as true lines in the edge map.

5 Bin size trade-off:

- Big bin: fewer votes, faster, lower accuracy
- Small bin: more votes, slower, higher accuracy

6 When the normal θ_n is known, we can narrow down the searching space from $\theta \in [0, 180]$ to $\theta \in [\theta_n - \Delta\theta, \theta_n + \Delta\theta]$.

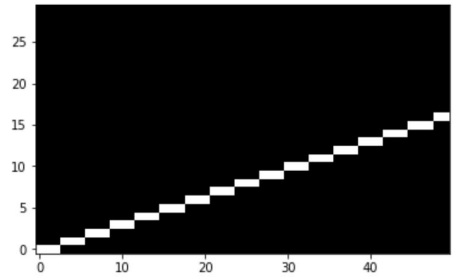
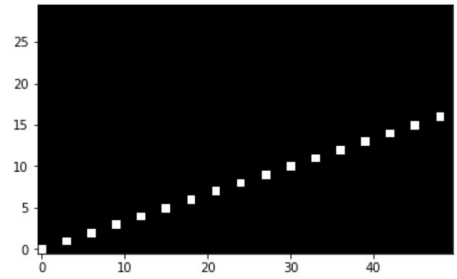
7 Three dimensions: circle center (a, b) and radius r

8 Given the implicit line equation: $x\cos(\theta) + y\sin(\theta) - d = 0$.

- For the nearly horizontal line ($\theta \in [45^\circ, 135^\circ]$),

$$
\begin{aligned}
&x\cos(\theta) + y\sin(\theta) - d = 0 \\
&y\sin(\theta) = -x\cos(\theta) + d \\
&\text{Since } \theta \in [45^\circ, 135^\circ],\ \sin(\theta) \neq 0 \\
&y = -\frac{\cos(\theta)}{\sin(\theta)}x + \frac{d}{\sin(\theta)}
\end{aligned}
$$

- The reason why we need to scan x and compute y for the nearly horizontal line:
  - It can avoid zero denominator. If scanning y and computing x, we may encounter $\cos(90^\circ) = 0$ on the denominator.
  - For the nearly horizontal line, the absolute slope of the line is smaller than 1. If we scan y (increment for the y-values is one), the x value of the line will change by more than one which leads to the gaps in the representation of the line in raster graphics.
  - Example: $y = \frac{1}{3}x$ (nearly horizontal line).

Scan x and compute y	Scan y and compute x
	

i. Similar to **1.h**

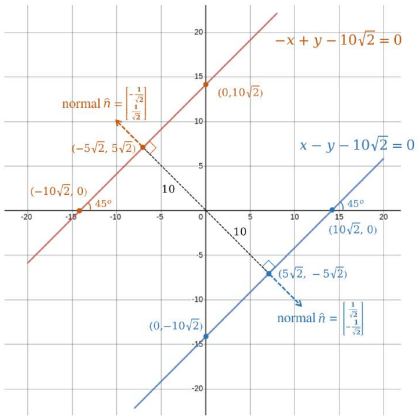
2. Model fitting

1 Answer:

- With $y = ax + b$, we cannot model vertical or near vertical lines because the slope would have to be infinite. This shows that we need to be careful when choosing the model so that it can describe all possible (and not only a subset) of observations.
- Vertical and near vertical lines.

2 Lines meet the requirements: $-x + y - 10\sqrt{2} = 0$ and $x - y - 10\sqrt{2} = 0$ $d = \frac{|c|}{\sqrt{a^2+b^2}} \Rightarrow 10 = \frac{|c|}{\sqrt{\quad}}$

- Parameters
  - Line 1: $a = -1, b = 1$, and $c = -10\sqrt{2}$
  - Line 2: $a = 1, b = -1$, and $c = -10\sqrt{2}$
- Draw and points on the line:

draw	points on line
	Line 1: $-x + y - 10\sqrt{2} = 0$ Points: $(-10\sqrt{2}, 0)$ and $(0, 10\sqrt{2})$ Line 2: $x - y - 10\sqrt{2} = 0$ Points: $(10\sqrt{2}, 0)$ and $(0, -10\sqrt{2})$

3 Answer:

- Implicit equation: $-x + y = 0$ or $x - y = 0$
- Normalized normal vector: $[-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^\top$ or $[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]^\top$

4 $l = [\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}}, -2]^\top$

e. -2.5

f. Answer:

- To fit a line using the implicit line equation: $l^\top p_i = 0$
  1. Build the correlation matrix $S = \sum_i p_i p_i^\top$.
  2. Find the eigenvector of S belonging to zero eigenvalue.
- The equation needs to solve:

$$\begin{cases} E(l) = l^\top S l \\ l^* = \arg\min_l E(l), \end{cases} \quad \text{where } S = \begin{bmatrix} \sum x_i^2 & \Sigma x_i y_i & \Sigma x_i \\ \sum x_i y_i & \Sigma y_i^2 & \Sigma y_i \\ \sum x_i & \sum y_i & n \end{bmatrix}$$

  Let $\nabla E(l) = 0$, so the equation that has to be solved is $Sl = 0$. The solution l^* is the eigenvector of S belonging to zero eigenvalue.

g. $S = \sum_{i=1}^n p_i p_i^\top = D^\top D$, where $D = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}$. Therefore, $S = \begin{bmatrix} 5 & 15 & 3 \\ 15 & 46 & 10 \\ 3 & 10 & 3 \end{bmatrix}$.

h. When a point p is off the curve f,

- the algebraic distance is: $d(p, f) = |f(p)|$.
- the geometric distance is: $d(p, f) = |p - x^*| = \frac{|f(p)|}{|\nabla f(x^*)|}$, where x^* is the closest point on the curve f.

i. Answer:

- The geometric distance:
  - Exact way: $d(p, f) = |p - x^*| = \frac{|f(p)|}{|\nabla f(x^*)|}$
  - Approximated: $d(p, f) \approx \frac{|f(p)|}{|\nabla f(p)|}$
- The reason for approximating is we cannot compute the exact value of x^*.

j. Algebraic distance: $d(p, f) = |f(p)| = 1$

k. Approximated geometric distance: $d(p, f) \approx \frac{|f(p)|}{|\nabla f(p)|} = \frac{1}{2}$

l. Answer:

- The objective function for active contours:

$$E(\phi(s)) = \int_{\phi(s)} (\underbrace{\alpha(s) E_{\text{cont}} + \beta(s) E_{\text{curv}}}_{\text{internal energy}} + \underbrace{\gamma(s) E_{\text{img}}}_{\text{external energy}}) ds,$$

  where $\alpha(s)$, $\beta(s)$, and $\gamma(s)$ are coefficients of the different energy terms.
- Components:
  - Continuity energy describes the contour behavior regarding elasticity or smoothness:

    $E_{\text{cont}} = |\frac{d\phi}{ds}|^2$
  - Curvature energy describes the contour behavior regarding curvature:

$E_{\text{curv}} = |\frac{d^2\phi}{ds^2}|^2$

- Image energy describes how the deformable curve will match with objects of the image: $E_{\text{img}} = -|\nabla I|^2$.

m. $p_1 = (1, 2)$, $p_2 = (2, 3)$, and $p_3 = (3, 4)$. At point p_2,

- $E_{\text{cont}} = |p_3 - p_2|^2 = |(1, 1)|^2 = (\sqrt{1^2 + 1^2})^2 = 2$
- $E_{\text{curv}} = |(p_3 - p_2) - (p_2 - p_1)|^2 = |(1, 1) - (1, 1)|^2 = 0$

n. We can set β to zero at corners to allow discontinuity.

3. Robust estimation

a. Answer:

- Outliers are the observation points that are distant from other observations.
- The fundamental problem is that the model fitting can be mismatched by the influence of outliers.

b. Answer:

- $E(\theta) = \sum_{i=1}^{n} \xi_\sigma(d(x_i; \theta))$
- In robust estimation, the function gives low weight for high-value outlier, however, the least squares objective function gives higher weight for high-value outlier.

c. Answer:

- Geman-McClure function: $\xi_\sigma(x) = \frac{x^2}{x^2+\sigma^2}$
- Advantages:
  - We can control the loss function.
  - The weight of outliers is up to 1 (has upper bound).
- Start with large σ and decrease as converging: $\sigma^{(n)} = 1.5 \times \text{median}\{d(x_i; \theta^{(n-1)})\}$

d. Geman-McClure function: $\xi_\sigma(x = 1, \sigma = 1) = \frac{x^2}{x^2+\sigma^2} = \frac{1}{2}$

e. Answer:

- RANSAC algorithm estimates the model parameters by repeated random sampling of observed data (the observed data contains both inliers and outliers). RANSAC uses the consensus scheme to find the best result. The principle of RANSAC algorithm:
  - Perform multiple experiments
  - Choose the best result
  - Use small sets in hope that at least one set will not have outliers
- The number of points drawn at each attempt should be small.
- A small number of points drawn can avoid including more outliers.

f. Answer:

- Parameters of RANSAC algorithm:
  - n : the number of points at each evaluation.

    - d : the minimum number of points needed.
    - t : the threshold to identify inliers.
    - k : the number of trials.
- The number of trials: $k = \frac{\log(1-p)}{\log(1-w^n)}$, where w is the probability that a data point is an inlier and p is at least one of the draws is free from outliers.

g. Number of experiments: $k = \frac{\log(0.01)}{\log(1-0.9^n)}$

n	k	n	k	n	k
2	2.7729774493739905	3	3.5271458279290657	4	4.314363283401872
5	5.15815564748312	6	6.074675084464547	7	7.077727323270934
8	8.18058587666166	9	9.396838847904254	10	10.740876345462974

Peoblem Set 2 Answerrs

1. Convolution layers

a. $\mathrm{R}:\begin{bmatrix}9 & 9\\ 9 & 9\end{bmatrix}$ $\mathrm{G}:\begin{bmatrix}18 & 18\\ 18 & 18\end{bmatrix}$ $\mathrm{B}:\begin{bmatrix}18 & 18\\ 27 & 27\end{bmatrix}$ $\xrightarrow{R+G+B}$ $\mathrm{Final}:\begin{bmatrix}45 & 45\\ 54 & 54\end{bmatrix}$

b. Answer:

$$\mathrm{R}:\begin{bmatrix}4 & 6 & 6 & 4\\ 6 & 9 & 9 & 6\\ 6 & 9 & 9 & 6\\ 4 & 6 & 6 & 4\end{bmatrix} \quad \mathrm{G}:\begin{bmatrix}8 & 12 & 12 & 8\\ 12 & 18 & 18 & 12\\ 12 & 18 & 18 & 12\\ 8 & 12 & 12 & 8\end{bmatrix} \quad \mathrm{B}:\begin{bmatrix}6 & 9 & 9 & 6\\ 12 & 18 & 18 & 12\\ 18 & 27 & 27 & 18\\ 14 & 21 & 21 & 14\end{bmatrix}$$

$$\xrightarrow{R+G+B} \mathrm{Final}:\begin{bmatrix}18 & 27 & 27 & 18\\ 30 & 45 & 45 & 30\\ 36 & 54 & 54 & 36\\ 26 & 39 & 39 & 26\end{bmatrix}$$

c. Answer:

$$\mathrm{R}:\begin{bmatrix}4 & 4 & 4 & 4\\ 4 & 4 & 4 & 4\\ 4 & 4 & 4 & 4\\ 4 & 4 & 4 & 4\end{bmatrix} \quad \mathrm{G}:\begin{bmatrix}8 & 8 & 8 & 8\\ 8 & 8 & 8 & 8\\ 8 & 8 & 8 & 8\\ 8 & 8 & 8 & 8\end{bmatrix} \quad \mathrm{B}:\begin{bmatrix}8 & 8 & 8 & 8\\ 12 & 12 & 12 & 12\\ 8 & 8 & 8 & 8\\ 12 & 12 & 12 & 12\end{bmatrix}$$

$$\xrightarrow{R+B+G} \mathrm{Final}:\begin{bmatrix}20 & 20 & 20 & 20\\ 24 & 24 & 24 & 24\\ 20 & 20 & 20 & 20\\ 24 & 24 & 24 & 24\end{bmatrix}$$

Also accept: $\mathrm{R}:\begin{bmatrix}4 & 4\\ 4 & 4\end{bmatrix}$ $\mathrm{G}:\begin{bmatrix}8 & 8\\ 8 & 8\end{bmatrix}$ $\mathrm{B}:\begin{bmatrix}12 & 12\\ 8 & 8\end{bmatrix}$ $\xrightarrow{R+B+G}$ $\mathrm{Final}:\begin{bmatrix}24 & 24\\ 20 & 20\end{bmatrix}$

d. When applying convolution, we do the dot product between the filter and the image. When the filter resembles the image, we expect a high response. The network is trying to find matches in the image.

e. When pooling between layers (or using convolution with a stride greater than 1), the spatial dimensions are sampled and so we get a pyramid with different spatial resolutions at the different layers. In this way, a fixed-size convolution filter covers a larger spatial region in deeper layers.

f. Answer:

- Increase the number of filters layers by layers.
- We decrease the spatial dimensions and increase the depth to keep the same number of coefficients. Also, it helps in learning more levels of global abstract structures and shrinking the feature space for input to the dense (fully connected) networks.

g. For output width, $W_{out} = \lfloor (W_{in} + 2P - F)/S + 1 \rfloor = 126$. Same for height, the final output shape is $126 \times 126 \times 16$

h. For output width, $W_{out} = \lfloor (W_{in} + 2P - F)/S + 1 \rfloor = 63$. Same for height, the final output shape is $63 \times 63 \times 16$

g. & h. Complete equation: Input $H_{in} \times W_{in} \times C_{in}$, convolved with C_{out} filters of size $\text{kernel\_size}\,[0] \times \text{kernel\_size}\,[1] \times C_{in}$, the output is $H_{out} \times W_{out} \times C_{out}$, where

$$H_{\text{out}} = \left\lfloor \frac{H_{\text{in}} + 2 \times \text{ padding}\,[0] - \text{dilation}[0] \times (\text{ kernel\_size}\,[0] - 1) - 1}{\text{stride}[0]} + 1 \right\rfloor$$

$$W_{\text{out}} = \left\lfloor \frac{W_{\text{in}} + 2 \times \text{ padding}\,[1] - \text{dilation}[1] \times (\text{ kernel\_size}\,[1] - 1) - 1}{\text{stride}[1]} + 1 \right\rfloor$$

i. Given an input feature with m channels, n one-by-one filters can be applied to reduce the number of channels from m to n ($m > n$) without changing spatial dimension.

j. Answer:

- The convolutional layers are used to extract image patterns or templates.
- Difference between early and deeper convolutional layers:
  - Early layers: extract simple patterns such as edge.
  - Deeper layers: extract complex patterns.

k. $R: \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ $G: \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$ $B: \begin{bmatrix} 2 & 2 \\ 4 & 4 \end{bmatrix}$

l. Downsampling the spatial dimension

m. Answer:

- The purpose of data augmentation is for better generalization and addresses the problem of limited data. It adds examples with perturbations (e.g. rotation, flip, contrast change) increasing variability in the training data which results in better generalization and prevents overfitting.
- It is most useful when the dataset is small.

2. CNNs

a. Answer:

- The goal of transfer learning is to improve learning in the target task by leveraging knowledge from the related source task.
- It is most useful when the target data is similar to the source data and there is significantly more data in the source dataset.

b. Avoid weights destruction by the gradient from untrained fully connected layers.

c. The coefficients of a pre-trained network can be fine-tuned by:

1. Add customized layers on top of the pre-trained layer.
2. Freeze the pre-trained layer.
3. Train the customized layer.
4. Unfreeze the pre-trained layer.
5. Train the entire network.

d. Answer:

- The inception blocks having filters with multiple receptive fields operate on the same level which can detect different size variations in the location of the information.
- GoogleNet uses auxiliary classifiers for addressing vanishing gradients.

e. Answer:

- Advantages of residual blocks:
  - With zero weights, the network computes the identity and can learn to zero blocks to eliminate unneeded layers.
  - Identity connections provide useful feedback throughout the network.
  - While increasing network depth, it avoids negative outcomes.
- Skip connections resolve the vanishing gradient problem. When the gradient passes through layers, it will become smaller. With residual blocks, the gradient can be passed directly through the skip connection and won't be smaller.

f. Answer:

- DenseNet uses direct connections (concatenation) from any layer to all subsequent layers, allowing for feature reuse and reducing the number of parameters.
- The complexity is controlled by
  - The growth rate k (the number of channels) so that each layer only produces k output feature maps.
  - Bottleneck layers are used to reduce the number of feature maps before each 3×3 convolutional layer in dense blocks, further reducing the number of parameters.
  - Transition layers are used between each dense block to further compact the model by reducing the number of feature maps and spatial dimensions.
  - Global Average Pooling (GAP) is used to replace the fully connected layer at the end of the network, reducing the number of parameters and improving regularization.

g. Given a 4×4 image with three channels where the first has all 1s, the second has all 2s and the third has all 3s, standard convolution with a 3×3 filter having all 1s in its first layer, all 2s in its second layer and all 3s in its third layer, depth-wise convolution with a 3×3 filter having all 1s in its first layer, all 2s in its second layer and all 3s in its third layer, and point-wise convolution with a filter with all 1s in it.

- For standard convolution, the output is $\begin{bmatrix} 126 & 126 \\ 126 & 126 \end{bmatrix}$.
- For depth-wise separable convolution:
  - The output of depth-wise convolution has three channels: the first is $\begin{bmatrix} 9 & 9 \\ 9 & 9 \end{bmatrix}$, the second is $\begin{bmatrix} 36 & 36 \\ 36 & 36 \end{bmatrix}$, and the third is $\begin{bmatrix} 81 & 81 \\ 81 & 81 \end{bmatrix}$.
  - Then, the output is convolved with the point-wise convolution (standard 1×1 convolution), and the result is $\begin{bmatrix} 126 & 126 \\ 126 & 126 \end{bmatrix}$.

h. MobileNets make computation faster by:

- Depth-wise separable convolution: the depth-wise convolution reduces the number of computations required in comparison to traditional convolution layers by applying a single filter

to each input channel. Then, the point-wise convolution combines the intermediate feature maps with a 1×1 filter, which greatly reduces the number of parameters.

- Width multiplier (fewer channels): reduce the number of input and output channels to each layer by a factor of $\alpha \in [0, 1]$.
- Resolution multiplier (smaller resolution): reduce the resolution by a factor $\rho \in [0, 1]$.

3. Object detection

a. The goal of object detection is to detect all instances of the predefined classes and provide its coarse localization in the image by axis-aligned boxes.

- Classification: classify the type of object in each bounding box. (classification problem)
- Localization: find the bounding box of different objects. (regression problem)

b. Answer:

- $\text{IoU} = \frac{|\text{pred} \cap \text{truth}|}{|\text{pred} \cup \text{truth}|} = \frac{16}{25+25-16} = \frac{16}{34}$
- $\text{Jaccard distance} = 1 - \text{IoU} = 1 - \frac{16}{34} = \frac{18}{34}$

c. r_i is recall at threshold t_i and p_i is precision at threshold t_i.

$AP_{0.5}$ (average precision) for the following precision-recall pairs: $(1, 0), (1, 0.2), (0.6, 0.4), (0.6, 0.6), (0, 0.8), (0, 1)$.

$$
\begin{aligned}
AP_{0.5} &= \sum_{\text{confidence threshold } t_i} (r_{i+1} - r_i) p_{i+1} \\
&= (0.2 - 0) \times 1 + (0.4 - 0.2) \times 0.6 + (0.6 - 0.4) \times 0.6 + (0.8 - 0) \times 0 + (1 - 0.8) \times 0 \\
&= 0.2 + 0.12 + 0.12 + 0 + 0 \\
&= 0.44
\end{aligned}
$$

d. Because we process the image at different scales, absolute coordinates cannot be used in so we normalize the box coordinates in $[0, 1]$.

e. Given the prediction $\hat{y} = [\hat{p}, \hat{x}_c, \hat{y}_c, \hat{w}, \hat{h}, \hat{c}_1, \ldots, \hat{c}_k] = [\, \hat{p}, \hat{\text{box}}, \hat{\text{class}} \,]$ and the groundtruth label $y = [p, x_c, y_c, w, h, c_1, \ldots, c_k] = [\, p, \text{box}, \text{class} \,]$, the general loss for object detection is $L = L_{reg} + L_{cls}$, where L_{reg} denotes the regression loss, e.g. L_2 loss, used to learn the bounding box and L_{cls} denotes the classification loss, e.g. cross-entropy, used to learn object type.

- If there is an object ($p = 1$), the loss $L = (p - \hat{p})^2 + L_{reg}(\text{box}, \hat{\text{box}}) + L_{cls}(\text{class}, \hat{\text{class}})$.
- If there is no object ($p = 0$), the loss $L = (p - \hat{p})^2$.

f. At each cell location, the output tensor shape is $10 \times (4 + 1 + K)$ for the 10 detection boxes, 4 bounding box offsets, 1 objectness prediction, and K class predictions.

g. Answer:

- Single-shot detectors perform object detection in a single network that looks at the image once.
- Two-shot detectors look at the image twice: one to propose object regions (region proposals) and a second to refine and classify regions. Two-shot detectors have a higher computational cost but may be more accurate.

h. The loss function of the YOLOv1:

$$
\begin{aligned}
L_{\text{YOLOv1}} = & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 \right] && (1) \\
& + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right] && (2) \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 && (3) \\
& + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2 && (4) \\
& + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{ classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2, && (5^*)
\end{aligned}
$$

where $\mathbb{1}_i^{\text{obj}}$ denotes if an object appears in cell i and $\mathbb{1}_{ij}^{\text{obj}}$ denotes that the jth bounding box predictor in cell i is responsible for that prediction. λ_{coord} increases the loss from bounding box coordinate predictions and λ_{noobj} decreases the loss from confidence predictions for boxes that don't contain objects.

(1) Center regression, $\mathbb{1}_{ij}^{\text{obj}} = 1$ if the jth bounding box predictor in cell i is responsible for that prediction.

(2) Height and width regression, $\mathbb{1}_{ij}^{\text{obj}} = 1$ if the jth bounding box predictor in cell i is responsible for that prediction.

(3) Object classification (Objectness loss), $\mathbb{1}_{ij}^{\text{obj}} = 1$ if the jth bounding box predictor in cell i is responsible for that prediction. The C denotes the confidence/objectness score.

(4) Non-object classification (Objectness loss), $\mathbb{1}_{ij}^{\text{noobj}} = 1$ if the jth bounding box predictor in cell i is not responsible for that prediction.

(5^*) Class prediction, $\mathbb{1}_{i}^{\text{obj}} = 1$ if an object appears in grid cell i. This term is for YOLOv1, which is different from later YOLO versions. In YOLOv1, the bounding box predictors are not responsible for class classification. Each bounding box consists of only 5 predictions: x, y, w, h, and confidence (C). The class prediction is made at the grid cell level. From YOLO9000 (YOLOv2), the class prediction is moved from the grid cell level to the bounding box level which means each bounding box consists of $5 + K$ predictions where K is the number of classes. And, this term is written as $\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \sum_{c \in \text{ classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2$.

Same as **3.e.**, if there is an object, the loss is the sum of objectness, regression, and class prediction loss, otherwise it only counts the objectness loss.

i. Answer:

- Region of Interest (RoI) pooling can be done by adjusting the window that is used to scan the input. It uses max pooling to convert the RoI to a fixed size to the Fully Connected (FC) classifier. RoI max-pooling works by dividing the $h \times w$ RoI window into an $H \times W$ grid of sub-windows of approximate size $h/H \times w/W$ and then max-pooling the values in each sub-window into the corresponding output grid cell.

- The FC classifier accepts fixed-size input, but the size of RoI may vary.

j. Answer:

- Select a detection in the grid cell with the highest detection score (probability) and delete detections that have an IoU (with the selected detection) that is greater than a threshold.
- We may detect the same object multiple times. Non-maximal suppression can be used to fix these multiple detections.

k. The general loss used in Mask R-CNN: $L = L_{cls} + L_{reg} + L_{seg}$

- Loss for classification task: $L_{cls} = L_{cls}^{RPN} + L_{cls}^{obj}$, where
  - L_{cls}^{RPN} is the (binary cross-entropy) loss for Region Proposal Network (RPN).
  - L_{cls}^{obj} is the (cross-entropy) loss for the object class prediction.
- Loss for regression task: $L_{reg} = L_{reg}^{RPN} + L_{reg}^{obj}$, where
  - L_{reg}^{RPN} is the (smooth L_1) loss for RPN.
  - L_{reg}^{obj} is the (smooth L_1) loss for the object.
  - We only compute regression loss when there is an object.
- Loss for segmentation task:
  - L_{seg} is (binary cross-entropy) loss for the segmentation.
  - We only compute segmentation loss when there is an object.

4. Semantic segmentation

a. Answer:

- Semantic segmentation classifies each pixel in an image into a class or object. The goal is to produce a dense pixel-wise segmentation map of an image, where each pixel is assigned to a specific class or object.
- Instance Segmentation involves identifying and separating individual objects within an image. The goal of instance segmentation is to produce a pixel-wise segmentation map of the image, where each pixel is assigned to a specific object instance.

b. Given a 5×5 image (x) and a 3×3 filter, the output (y) size is 3×3 (without padding). After vectorization, the image (x) is a $25D$ vector and the output (y) is a $9D$ vector, therefore the size of the matrix (F) is 9×25 ($y_{\underline{9 \times 1}} = F_{\underline{9 \times 25}}\ x_{\underline{25 \times 1}}$).

c. The size of the transpose convolution matrix is 25×9.

d. Answer:

- We lose spatial resolution in the encoder which makes it hard to decode accurately. Therefore, we want to mix the features from both the encoder and decoder.
- The skip connections allow us to concatenate the encoder features with the corresponding decoder features so that the high-resolution features from the encoder can be used in the decoder to assemble a more precise output.

e. DeepLab has an encoder-decoder architecture and uses the Atrous Spatial Pyramid Pooling (ASPP) with skip connection. The convolutions are implemented as depth-wise separable convolutions.

f. The most common performance metric used in semantic segmentation is the mean IoU (mIoU) which is defined as the average IoU over all classes.

Problem Set 3 Answers

1. Camera Calibration 1

a. Answer:

- Forward projection: Given world point $\underline{P}_i$ and project matrix M, compute image point P_i
- Calibration: Given world point $\underline{P}_i$ and image point P_i, compute project matrix M
- Reconstruction: Given image point P_i and project matrix M, compute the world point $\underline{P}_i$
- Forward projection is the easiest.
- Reconstruction is the hardest.

b. The world coordinate of point and corresponding image coordinate of point

c. Steps of non-planar calibration:

1. Estimate projection matrix M
2. Find parameters: K^*, R^*, T^*

d. Given world point $\underline{P}_i = [1, 2, 3]^T$ and project matrix M, the image point $P_i = M\underline{P}_i$

$$P_i = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 18 \\ 14 \\ 7 \end{bmatrix}_{\text{2DH}} \Rightarrow \begin{bmatrix} \frac{18}{7} \\ 2 \end{bmatrix}_{\text{2D}}$$

e.

$$\begin{bmatrix} 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & -100 & -200 & -300 & -100 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & -200 & -400 & -600 & -200 \end{bmatrix}$$

f. Answer:

- 6 non-coplanar points for non-planar calibration and 4 non-collinear points for planar calibration
- Solution obtained by:
  1. Build the matrix A for $AX = 0$ from world point and image point
  2. Use SVD to decompose the matrix $A = U\Sigma V^T$, and the last column of matrix V is the $\hat{M}$
  3. $M = \xi\hat{M}$, where $|\xi| = 1/|a_3|$ and a_3 is the last row of matrix $\hat{M}$ except last column element

g. The principal is that $M = K^*[R^*|T^*] = \xi\hat{M}$, where R^* is orthogonal matrix which means the row of vector in R^* : r_1, r_2, r_3 is orthogonal to each other and is unit vector. So we can use dot product: $r_1 \cdot r_2 = 0, r_2 \cdot r_3 = 0$, and $r_1 \cdot r_3 = 0$ and cross product: $r_1 \times r_2 = r_3, r_1 \times r_3 = r_2$, and $r_2 \times r_3 = r_1$ to compute the parameters.

h. Answer:

$$\text{MSE: } E(K^*, R^*, T^*) = \frac{1}{n}\sum_{i=1}^{n}\left((x_i - \frac{m_1^T\underline{P}_i}{m_3^T\underline{P}_i})^2 + (y_i - \frac{m_2^T\underline{P}_i}{m_3^T\underline{P}_i})^2\right)$$

i. Answer:

- The principle of planar camera calibration:

1. Estimate the 2D homography matrix between calibration target and image.
    1. The coordinate of z of world point is 0, so estimate the 2D homography matrix H.
    2. Build the matrix A for $AX = 0$ from world point and image point.
    3. Use SVD to decompose the matrix $A = U\Sigma V^T$, and the last column of matrix V is the $\hat{H}$.
    4. Then we assume $H = \alpha\hat{H}$
2. Estimate intrinsic parameters from several views.
    1. According to $H = \alpha\hat{H}$, and the rotation vector r_1 and r_2 are orthogonal to each other, we get two dot product equations:

    $$\begin{cases} r_1 \cdot r_1 = r_2 \cdot r_2 = 1 \Rightarrow \boxed{\hat{h}_1^T S \hat{h}_1 = \hat{h}_2^T S \hat{h}_2} \\ r_1 \cdot r_2 = 0 \Rightarrow \boxed{\hat{h}_1^T S \hat{h}_2 = 0} \end{cases}, \text{where } S = K^{*-T} K^{*-1}$$

    2. We need at least 3 homography matrices (3 views), then we can build 6 equations to solve the K^* with 5 unknown parameters.
    3. Build the matrix B for $BS = 0$ using several views. Use SVD to decompose the matrix $B = U\Sigma V^T$, and the solution is the last column of matrix V.
    4. Compute intrinsic parameters by equations.
3. Compute extrinsic parameters for any views. According to $H = \alpha\hat{H}$, we can get:

$$r_1 = \alpha K^{*-1}\hat{h}_1 \quad \text{and} \quad r_2 = \alpha K^{*-1}\hat{h}_2 \quad \text{and} \quad T^* = \alpha K^{*-1}\hat{h}_3$$

- All points in same planar in planar calibration, while it is not in non-planar calibration.

j. Answer:

- The difference is that there are two rotation vectors r_1 and r_2 in homography matrix,while there are three rotation vectors r_1, r_2, and r_3 in projection matrix.
- We assume the z coordinate is 0 in world point $\underline{P}_i$

2. Camera Calibration 2

a. $\begin{bmatrix} 3 & 4 & 5 & 1 & 0 & 0 & 0 & 0 & -3 & -4 & -5 & -1 \\ 0 & 0 & 0 & 0 & 3 & 4 & 5 & 1 & -6 & -8 & -10 & -2 \end{bmatrix}$

b. $M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \end{bmatrix} = \left[\begin{array}{c|c} -a_1^T- & \\ -a_2^T- & b \\ -a_3^T- & \end{array}\right]$

$m_1 = [1,2,3,4]$
$m_2 = [2,3,4,5]$
$m_3 = [3,4,5,6]$

$u = \frac{m_1 \cdot m_3}{m_3 \cdot m_3} = \frac{3+8+15}{9+16+25} = \frac{26}{50}$

$v = \frac{m_2 \cdot m_3}{m_3 \cdot m_3} = \frac{6+12+20}{9+16+25} = \frac{38}{50}$

- $u_0 = |\xi|^2 a_1 \cdot a_3 = \frac{a_1 \cdot a_3}{|a_3|^2} = \frac{3+8+15}{9+16+25} = \frac{26}{50} = 0.52$
- $v_0 = |\xi|^2 a_2 \cdot a_3 = \frac{a_2 \cdot a_3}{|a_3|^2} = \frac{6+12+20}{9+16+25} = \frac{38}{50} = 0.76$

c. $M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \end{bmatrix}$, $p = [1,2]^T$, and $\underline{P} = [3,4,5]^T \Rightarrow [3,4,5,1]^T_{\text{3DH}}$

The projected point: $\hat{p} = M\underline{P} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 30 \\ 43 \\ 56 \end{bmatrix}_{\text{2DH}} \Rightarrow \begin{bmatrix} 30/56 \\ 43/56 \end{bmatrix}_{\text{2DH}}$

The projection error: $MSE = |p - \hat{p}|^2 = (1 - 30/56)^2 + (2 - 43/56)^2 \approx 1.7337$

d. $I + Q = \left[\begin{array}{c|c} R^*_{3\times 3} & \mathbf{0}_{3\times 1} \\ \hline \mathbf{0}_{1\times 3} & 1 \end{array}\right]_{\text{3DH}} = \left[\begin{array}{ccc|c} 6 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]_{\text{3DH}}$

- $R = (R^*)^T = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\text{3D}}$ =
- $T = -RT^* = -(R^*)^T T^* = [-6 \quad -2 \quad -3]^T_{\text{3D}}$ = 

e. $\begin{bmatrix} 3 & 4 & 1 & 0 & 0 & 0 & -3 & -4 & -1 \\ 0 & 0 & 0 & 3 & 4 & 1 & -6 & -8 & -2 \end{bmatrix}$

P = (3,4,0)
P = (1,2)

3. Multiple View Geometry 1

a. Answer:

- Sparse stereo matches specific points and dense stereo matches each pixel.
- Sparse can be used for far (between frames) views, while dense should be used for close (between frames) views. Both cannot match uniform patch, invisible in one view, and ambiguity cases.

b. Answer:

- NCC computes the normalized cross correlation of two windows around the point. SSD computes the sum of square distance between two windows around the point. They are used to compute similarity to determine correspondence.
- If allowing the search space to be the entire image, we might get more mismatching points as we are more likely to get errors.
- The solution is to compute the epipole line.

c. $z = f\frac{T}{d} = 10 \times 100/(x_r - x_l) = 1000/3$

d. The point pair may not match correctly and incorrect matching will lead to incorrect depth recovery.

3.5 Given R_l, T_l (rotation and translation of the left camera with respect to the world) and R_r, T_r (rotation and translation of the right camera with respect to the world), write the expression for the rotation and translation of the right camera with respect to the left camera.

- $R = R_l^T R_r$
- $T = R_l^T (T_r - T_l)$

4. Multiple View Geometry 2

a. $z = f\frac{T}{d} = 20/3$

b. $\begin{bmatrix} 0 & -3 & 2 \\ 3 & 0 & -1 \\ -2 & 1 & 0 \end{bmatrix}$

$\lceil 1 \quad 2 \quad 3 \rceil \lceil 1 \rceil \qquad\qquad \lceil 1 \rceil$

4.4 Given corresponding left and right points $(1, 2)$ and $(2, 3)$ respectively, write the respective row in the matrix that must be formed to solve for the fundamental matrix.

d. The left point is $P_l = (x_l = 1, y_l = 2)$ and right point is $P_r = (x_r = 2, y_r = 3)$

$$P_r^T F P_l \Rightarrow [x_r x_l \quad x_r y_l \quad x_r \quad y_r x_l \quad y_r y_l \quad y_r \quad x_l \quad y_l \quad 1] = [2 \quad 4 \quad 2 \quad 3 \quad 6 \quad 3 \quad 1 \quad 2 \quad 1]$$

OR

$$P_l^T F P_r \Rightarrow [x_l x_r \quad x_l y_r \quad x_l \quad y_l x_r \quad y_l y_r \quad y_l \quad x_r \quad y_r \quad 1] = [2 \quad 3 \quad 1 \quad 4 \quad 6 \quad 2 \quad 2 \quad 3 \quad 1]$$

Problem Set 4 Answers

1. Stereo

1.1 Sparse stereo matches specific points, while dense stereo matches each pixel. Sparse stereo is suitable for far (between frames) views, while dense stereo should be used for close (between frames) views. Both approaches fail in cases of uniform patches, objects invisible in one view, and ambiguous matches.

1.2 Answer:

- **NCC (Normalized Cross-Correlation):**

$$\phi(w_1, w_2) = \sum_i \left(\frac{w_1(x_i, y_i) - u_1}{\sigma_1} \right) * \left(\frac{w_2(x_i, y_i) - u_2}{\sigma_2} \right)$$

- **SSD (Sum of Squared Differences):**

$$\phi(w_1, w_2) = \sum_i \left(w_1(x_i, y_i) - w_2(x_i, y_i) \right)^2$$

These metrics help measure similarity to determine correspondence.

- Risk of searching the entire image: Increases the chance of mismatches due to errors.
- Solution: Compute the epipolar line to restrict the search space.

1.3 The depth z of the 3D point is:

$$z = \frac{fT}{d} = \frac{10 \cdot 100}{3} = \frac{1000}{3}$$

1.4 Point pairs may not match correctly. Incorrect matching will lead to erroneous depth recovery.

1.5 Answer:

$$R = R_l^T R_r, \quad T = R_l^T (T_r - T_l)$$

2. Epipolar Geometry

2.1 The optical centers of the camera lenses are distinct, and each center projects onto a distinct point in the other camera's image plane. These points, denoted by e_l and e_r, are called epipoles.

The epipolar line is formed by the intersection of the image planes with the epipolar plane. The epipoles may be inside or outside the image.

2.2 The essential matrix E is:

$$E = R^T [T]_x$$

The epipolar constraint equation is:

$$P_r^T E P_l = 0$$

2.3 The fundamental matrix F is:

$$F = K_r^{-T} E K_l^{-1}$$

The epipolar constraint equation is:

$$\tilde{P_r}^T F \tilde{P_l} = 0$$

2.4 The rank of the essential and fundamental matrices is 2 because $[T]_x$ is a skew-symmetric matrix.

2.5 The corresponding right epipolar line is:

$$F P_l$$

2.6 The corresponding left epipolar line is:

$$F^T P_r$$

2.7 Find the matrix F directly from 8-point correspondences.

2.8 The first row of the matrix for the 8-point algorithm is:

$$[5000, 10000, 50, 10000, 20000, 100, 100, 200, 1]$$

2.9 To normalize the points:

$$q_i = \frac{P_i - u_p}{\sigma_p}, \quad q_i' = \frac{P_i' - u_p'}{\sigma_p'}$$

Normalization improves numerical stability and accuracy. The fundamental matrix of the original points is recovered as:

$$F = M^T F' M$$

2.10 The epipoles can be recovered from the fundamental matrix F:

$$F^T P_r = 0, \quad F = V D U^T$$

The right epipole is the last column of U, and the left epipole is the last column of V.

3. Reconstruction

3.1 Answer:

1. Align the right image with the left image.
2. Find a coordinate system aligned with the baseline.
3. Make the planes co-planar and scale using K.

After alignment, the epipolar lines become parallel.

3.2 Answer:

- Absolute reconstruction: Complete reconstruction with known intrinsic and extrinsic parameters.
- Euclidean reconstruction: Up to a known scale, with known intrinsic parameters.

- Projective reconstruction: Reconstruction up to an unknown 3D projective map.

3.3 The matrix A to solve for coefficients $ (a, c, b) $ is:

$$A = \begin{bmatrix} P_l \\ P_l \times RP_r \\ -RP_r \end{bmatrix}$$

Solve using:

$$[a, b, c]^T = A^{-1}T$$

3.4 Using the coefficients $ (a, b, c) $, the 3D point P is:

$$P = aP_l + 0.5cW = 0.5(aP_l + bRP_r + T)$$

3.5 There is an unknown scale in Euclidean reconstruction because the baseline is unknown. This scale can be resolved by measuring the distance between two points in image coordinates and the corresponding distance in world coordinates.

3.6 Answer:

$$E^T E = [T]_x^T [T]_x, \quad \text{tr}(E^T E) = 2\|T\|^2$$

We normalize E as:

$$\hat{E} = 2E/\text{tr}(E^T E)$$

The baseline in $\hat{E}$ has a length of 1.

3.7 Answer: There are four possible sign combinations of T, R:

- $(+, +)$
- $(+, -)$
- $(-, +)$
- $(-, -)$

We reconstruct according to each option and choose the one with all positive z-coordinates.

Problem Set 5 Answers

1. Motion

1.1 3D motion vectors represent motion in the real world. 2D projected motion vectors are the projection of 3D motion vectors onto the image plane. Optical flow vectors are the observed 2D motion vectors derived from image sequences.

Example: A uniformly colored rotating sphere in the real world will not produce optical flow vectors because there are no discernible image features to track.

1.2 The projected motion field in a video taken by a car driving on a straight road while looking to the side is a parallel motion field with vectors parallel to the car's motion direction. Motion vectors will be larger for world points closer to the car (i.e., smaller z-values).

1.3 The projected motion field in a video taken by an airplane aiming to land at a fixed point while looking forward is a radial motion field with a focus of expansion. Motion vectors will be larger for points farther from the instantaneous epipole and closer to the ground plane (smaller z-values).

1.4 The fundamental motion projection equation is:

$$v = \frac{f}{Z}(ZV - V_z P)$$

1.5 Assuming 3D motion with translational velocity τ and rotational velocity ω, write the equation for the projected translational and rotational motion.

1.5 For 3D motion with translational velocity τ and rotational velocity ω, the projected translational and rotational motions are:

- Motion components:

$$\begin{aligned} v_x &= v_x^{(\tau)} + v_x^{(\omega)} \\ v_y &= v_y^{(\tau)} + v_y^{(\omega)} \\ v_z &= 0 \end{aligned}$$

- Translational motion:

$$\begin{aligned} v_x^{(\tau)} &= \frac{\tau_z x - \tau_x f}{Z} \\ v_y^{(\tau)} &= \frac{\tau_z y - \tau_y f}{Z} \end{aligned}$$

- Rotational motion:

$$\begin{aligned} v_x^{(\omega)} &= -\omega_y f + \omega_z y + \frac{\omega_x x y}{f} - \frac{\omega_y x^2}{f} \\ v_y^{(\omega)} &= \omega_x f - \omega_z x - \frac{\omega_y x y}{f} + \frac{\omega_x y^2}{f} \end{aligned}$$

1.6

- Pure translational motion without z-motion: The projected motion field is parallel.
- Pure translational motion with z-motion: The projected motion field is radial with a focus of expansion or contraction.

1.7 The instantaneous epipole coordinates are:

$$x_0, y_0 = \left(\frac{f\tau_x}{\tau_z}, \frac{f\tau_y}{\tau_z} \right)$$

1.8 Motion parallax occurs when objects at different depths move relative to one another due to the observer's motion. The relative motion field equations are:

$$\Delta v_x = (x - x_0)\tau_z \left(\frac{1}{Z} - \frac{1}{Z'} \right)$$
$$\Delta v_y = (y - y_0)\tau_z \left(\frac{1}{Z} - \frac{1}{Z'} \right)$$

2. Optical Flow

2.1 The optical flow constraint equation (OFCE) is:

$$\nabla I \cdot v = -I_t$$

The basic assumption is that image brightness remains constant at each location of the object.

2.2 The aperture problem arises because the OFCE provides only the motion component projected onto the image gradient direction. From a single point, we can recover motion only in the direction of the gradient.

2.3 Block-based methods address the aperture problem by including multiple image gradients within a window. This provides projections of the motion vectors in multiple directions, enabling a complete motion vector estimate.

2.4 The objective function for block-based optical flow estimation is:

$$E(v) = \sum_w (\nabla I \cdot v + I_t)^2$$

The system of equations to solve for optical flow v is:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i) I_y(q_i) \\ \sum_i I_y(q_i) I_x(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i) I_t(q_i) \\ -\sum_i I_y(q_i) I_t(q_i) \end{bmatrix}$$

In weighted block methods, weights are assigned such that locations closer to the center of the window get higher weights. These weights modify the summations in the equations to emphasize central pixels.

2.5 Affine motion estimation does not assume constant optical flow in each window.

Objective:

$$E(\mathbf{a}) = \sum_{(x,y) \in \text{patch}} (I_x u + I_y v + I_t)^2$$

When substituting the affine model:

$$E(\mathbf{a}) = \sum_{(x,y)\in \text{patch}} \left(I_x(a_1 + a_2x + a_3y) + I_y(a_4 + a_5x + a_6y) + I_t\right)^2$$

Solution: Affine flow vectors are computed using the estimated affine parameters a. To compute a, solve:

$$\begin{bmatrix} \sum I_x^2 & \sum I_x^2 x & \sum I_x^2 y & \sum I_x I_y & \sum I_x I_y x & \sum I_x I_y y \\ \sum I_x^2 x & \sum I_x^2 x^2 & \sum I_x^2 xy & \sum I_x I_y x & \sum I_x I_y x^2 & \sum I_x I_y xy \\ \sum I_x^2 y & \sum I_x^2 xy & \sum I_x^2 y^2 & \sum I_x I_y y & \sum I_x I_y xy & \sum I_x I_y y^2 \\ \sum I_x I_y & \sum I_x I_y x & \sum I_x I_y y & \sum I_y^2 & \sum I_y^2 x & \sum I_y^2 y \\ \sum I_x I_y x & \sum I_x I_y x^2 & \sum I_x I_y xy & \sum I_y^2 x & \sum I_y^2 x^2 & \sum I_y^2 xy \\ \sum I_x I_y y & \sum I_x I_y xy & \sum I_x I_y y^2 & \sum I_y^2 y & \sum I_y^2 xy & \sum I_y^2 y^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} -\sum I_t I_x \\ -\sum I_t I_x x \\ -\sum I_t I_x y \\ -\sum I_t I_y \\ -\sum I_t I_y x \\ -\sum I_t I_y y \end{bmatrix}$$

The affine flow vecctors are then computed using:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_1 \\ a_4 \end{bmatrix} + \begin{bmatrix} a_2 & a_3 \\ a_5 & a_6 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$