Computer Vision - HW 4

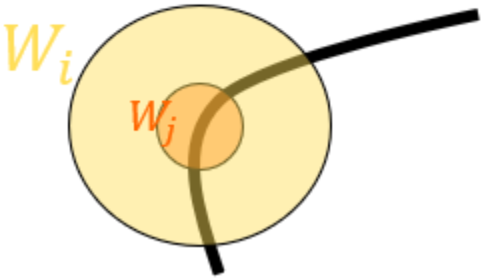
Tom Sabag 208845842
Gal Moshkovitch 315848929

Question 1

Assume that you apply the optical flow algorithm (OF) on a pair of images with a given set of parameters. Let p be a pixel for which the algorithm fails to compute the OF. Moreover, changing a single parameter results in computing the OF of p. Write what the parameter may be, why the algorithm fails in the first case and why it succeeds in the second case.

Answer 1

Parameter: Window Size
Explanation:
As learned in class, the Optical Flow algorithm is based on motion vectors $u(p), v(p)$ are estimated for each pixel based on a fixed size patch around its neighborhood. Flow estimation is gradient based, and is possible only if there's enough texture around p
(i.e. $Rank(C) = 2$).
Example
$p's$ optical flow calculation is based on a certain window size. Using a small W_j might fail due to lack of spatial information. In the following example, corners are misclassified as edges ($Rank(C) = 1$) because of an inadequate window size:



Enlarging the window size to W_i will result in a $Rank(C) = 2$.
As a result:
$C = A^T A$ is invertible and $A^+ = (A^T A)^{-1} A^T$ exists,

Plugging it into the Optical Flow equation:

$$A \cdot \left(u(p), v(p)\right)^T = b$$
$$\left(u(p), v(p)\right)^T = A^+ b$$

Conclusion: Inadequate window size W_j may fail the optical equation calculation. In the example above we showed how using a different window size W_i enables flow calculation for a pixel p.

Question 2

Consider the optical flow algorithm which we learned in class. On which camera motion it is expected to fail? Give a short explanation to your answer.

Answer 2

Optical flow algorithm is based on the underlying assumptions that motion is constant and smooth in $W(p)$. While small translations are captured well, other types of motions might be harder to capture:

- **Rotation:** pixels of different distances from the center of rotation move at different speeds and in different directions. As learned in class, we assume that $u(p), v(p)$ are small and constant in $W(p)$.
- **Large Translations:** Estimating $u(p), v(p)$ is based on gradients around $W(P)$:

  $$\underbrace{\begin{pmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_k) & I_y(p_k) \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} u(p_0) \\ v(p_0) \end{pmatrix} = -\underbrace{\begin{pmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_k) \end{pmatrix}}_{\mathbf{b}}$$

  Optical flow assumes local spatial coherence, i.e., that nearby pixels exhibit similar motion. Large translations might result in p_k not being in $p's$ neighborhood in adjacent frames. As a result, $u(p), v(p)$ cannot be estimated.
- **Scale:** When zooming in/out, adjacent pixels don't move like their neighbors. As a result, the assumption that optical flow vectors in a neighborhood are constant doesn't hold.

Question 3

Assume two pixels have the same optical flow and the camera is static. Does it necessarily implies that they are projections of two 3D points that move at the same 3D direction? If so explain why, and if not give a specific counter example**.**

Answer 3

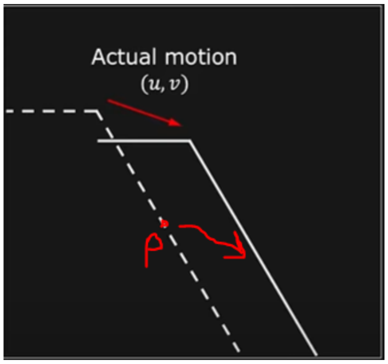
Same optical flow for two pixels does not necessarily mean that the 3D points they represent movement in the same direction due to the 'Aperture Problem'.

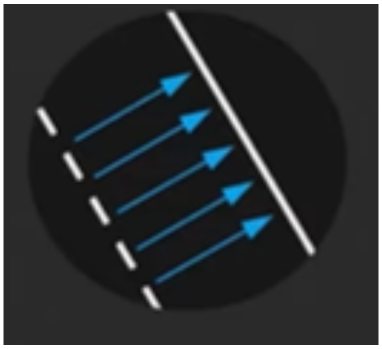
Aperture Problem

Optical flow algorithm uses a fixed size window W_i estimating motion of each pixel. Because only a small patch in the image is viewed at a time, as if it was viewed through a small aperture, the motion cannot be uniquely defined and may be ambiguous.

Example

Let p be a pixel on a straight line. Estimating visually the optical flow of p, without the limitations of aperture vision, the motion is directed towards the right and downward.
However, when the view is limited as a result of an aperture, the perceived motion is directed towards the right and upward instead.





Conclusion

Points in 3D can move in different directions but have the same optical flow because of the aperture problem.

Question 4

Consider two images of the same static scene that were taken from two cameras. Assume that the COP of the cameras are identical (no translation only rotation and maybe different internal parameters).

Prove formally: the two images are related by an homography transformation.

Hint: You can prove it using the assumption that the world coordinate system is the same as the coordinate system of one of the cameras, and the rotation between the cameras, as well as the internal parameters of each camera are known. If you use these assumptions, you have to explain why it is ok to use them.

Answer 4

To simplify the computation, we will align the world coordinate system with the cameras' coordinate system, by placing the origin at the center of the projection.

Let $M_{int}^1 = \begin{bmatrix} f_x^1 & 0 & c_x^1 \\ 0 & f_y^1 & c_y^1 \\ 0 & 0 & 1 \end{bmatrix}, M_{int}^2 = \begin{bmatrix} f_x^2 & 0 & c_x^2 \\ 0 & f_y^2 & c_y^2 \\ 0 & 0 & 1 \end{bmatrix}$ be the intrinsic matrices with internal parameters of the two cameras.

Let $p_1 = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ a pixel in the first camera in homogeneous coordinates.

Using M_{int}^1, we can project p_1 to a ray in the 3d space $\widetilde{P_1} = \left(M_{int}^1\right)^{-1} \cdot \widetilde{p_1} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$.

By assuming no translation and a common center of projection, we can simplify the extrinsic matrix between the cameras to $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$, the rotation matrix.

The ray in the coordinates of the second camera is $\widetilde{P_2} = R \cdot \widetilde{P_1}$.

Using M_{int}^2 we get the projection of the ray to the pixel of the second image $\widetilde{p_2} = M_{int}^2 \cdot \widetilde{P_2}$.

We get that $\widetilde{p_2} = M_{int}^2 \cdot R \cdot \left(M_{int}^1\right)^{-1} \cdot \widetilde{p_1} =$

$$\begin{bmatrix} f_x^2 & 0 & c_x^2 \\ 0 & f_y^2 & c_y^2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{f_x^1} & 0 & -\frac{c_x^1}{f_x^1} \\ 0 & \frac{1}{f_y^1} & \frac{-c_y^1}{f_y^1} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

$$H \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

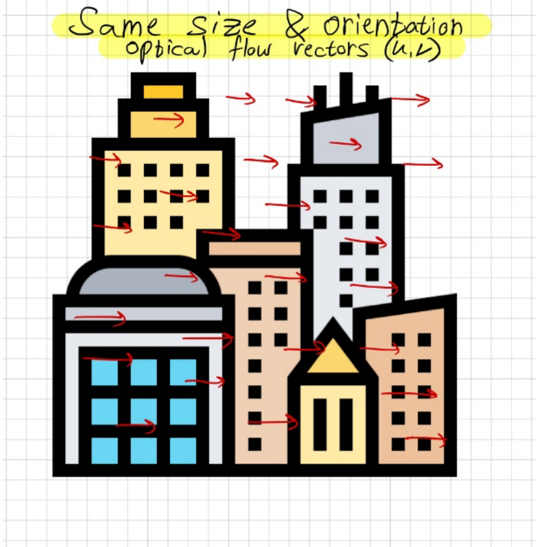
H is a 3x3 matrix that describes the mapping between the two planes.

Question 5

Consider a video captured by a camera that is attached to the side of a car that moves on a straight road. Assume you have a perfect optical flow algorithm, which is applied to the video. What is the expected optical flow (OF) of the projection of the buildings that are parallel to the road with the same distance to the road? Describe its orientation and size, and whether it is fixed for all pixels that are projection of these buildings. Give a short explanation for your answer.

Answer 5

Expected building OF:



Orientation: Horizontal for all pixels.
Explanation: W.l.o.g, assume that the camera is attached to the right side of the car. Since the buildings are parallel to the moving car, the relative motion of the buildings with respect to the car will be horizontal to the right (assuming no ups and downs).

Size: equal for all pixels.
Explanation: As taught in geometry lessons, when objects undergo horizontal displacement, closer objects appear to move at a faster pace, while those farther away appear to move more slowly. Since all buildings are located at the same distance from the driving car, their motion will be perceived at the same pace. Therefore, the optical flow vectors $(u(p), v(p))$ will be proportional to the distance of the buildings from the road (same for all pixels) and the speed of the car (same camera, therefore same for all pixels).

Question 6

Assume the OF we learned in class is applied once to a pair of successive frames and once to frames that are 20 frames apart using the same set of parameters. You may assume that the camera motion is constant, and that the scene is static.

(a) On which regions the computation of the OF is expected to fail in both cases? Give a short explanation to your answer, including algebraic justification.

(b) Where the OF is expected to fail only for the 20 frames apart case? Explain your answer and suggest a method to overcome this failure.

Answer 6

(a) Not enough texture

Under the BCA assumption, the intensity of an object point doesn't change between frames f_t, f_{t+dt}, therefore

$I\big(p_x + dx, p_y + dy, t + dt\big) = I(p_x, p_y, t)$

Additionally, we can estimate the object point pixel intensity using a Tailor Series approximation:

$I\big(p_x + dx, p_y + dy, t + dt\big) \cong I\left(p_x, p_y, t\right) + \frac{\partial I}{\partial x}(p)dx + \frac{\partial I}{\partial y}(p)dy + \frac{\partial I}{\partial t}(p)dt$

Comparing both sides, we concluded that:

$$\frac{\partial I}{\partial x}(p)dx + \frac{\partial I}{\partial y}(p)dy + \frac{\partial I}{\partial t}(p)dt = 0$$

Divide both sides by dt, we get:

$$\frac{\partial I}{\partial x}(p)\frac{dx}{dt} + \frac{\partial I}{\partial y}(p)\frac{dy}{dt} = -\frac{\partial I}{\partial t}(p)$$

Denote $u(p) = \frac{dx}{dt}$ the optical flow vector in the x axis, and

Denote $v(p) = \frac{dy}{dt}$ the optical flow vector in the y axis.

In matrix notation:

$$\big(I_x(p), I_y(p)\big) \cdot \big(u(p), v(p)\big)^T = -I_t(p)$$

Under the assumption of constant camera motion as mentioned above, neighboring pixels have the same optical flow vectors $u(p), v(p)$, therefore

$$\forall p_i \in W(p):$$
$$\big(I_x(p_i), I_y(p_i)\big) \cdot \big(u(p), v(p)\big)^T = -I_t(p_i)$$

Denote $A = \big(I_x(p_i), I_y(p_i)\big)$. $A \in R^{kx2}$, and in order to calculate the optical flow vectors we need to use its pseudo inverse A^+. As learned in class, $A^T \cdot A = C$, i.e. Harris' Matrix. We learned in previous classes that C is has a rank of 2 only of there's enough texture within $W(p)$.

Conclusion: calculating the optical flow gradients $u(p), v(p)$ requires an invertible C, which is only possible if enough texture around p exists.

(b) The optical flow algorithm is expected to fail only for the 20 frames apart case in regions where significant motion occurred, violating the assumption of local spatial coherence. As learned in class, in order to overcome large motions when calculating OF, **a pyramid of optical flows can be used**. Large motions in reduced resolution images appear to be smaller and might be captured in comparison to higher resolution images. Using multiple scales enable the OF algorithm to capture both smaller and larger motions.

Question 7

The basic change detection algorithm we learned in class, computes a single image as a background model using a median image. The algorithm has several parameters.

(a) List the set of parameters.

(b) Assume that the intensity values of the pixel p at frame i is given by p[i], and p = [10, 10, 100, 10, 202, 10, 30, 205, 201, 200, 201]. Using one set of parameters, the value p(11) = 201 was detected as background while using another set of parameters, the value p(11) = 201 was detected as foreground. Give a single parameter of the algorithm such that changing its value can explain this difference. Give a short explanation for your answer including the list of all parameters in the two cases.

Answer 7

(a) Parameters:

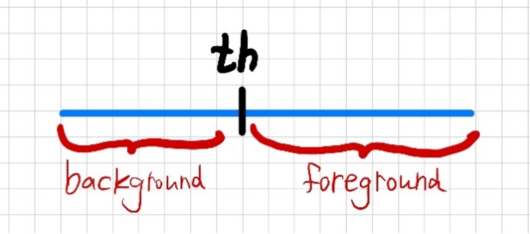
- $\boldsymbol{n}$**:** Amount of frames ($\boldsymbol{I_n}$)
- $\boldsymbol{th}$**:** a threshold determining whether a pixel is a background or foreground

(b) Parameter: th

Explanation:

Sorted Pixels & Median: [10, 10, 10, 10, 30, 100, 200, 201, 201, 202, 205].

In the basic change detection algorithm learned in class, the background $I_B(p)$ is the median pixel value p across I_n frames. If $|I_B(p) - I_k(p)| < th$ than p is classified as a background pixel.



Example:

- ○ $\boldsymbol{th = 100 => Foreground\ Classification}$
  $|I_B(p) - I_k(p)| = |100 - 205| = 105 > 100 = th$

- ○ $\boldsymbol{th = 110 => Background\ Classification}$
  $|I_B(p) - I_k(p)| = |110 - 205| = 95 < 100 = th$

Conclusion:

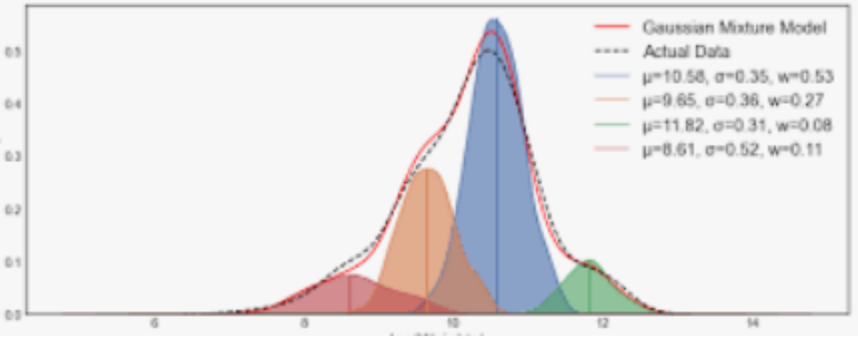
Changing the threshold parameter can influence the classification of pixels as background / foreground.

Question 8

Suggest a change detection algorithm where the background value of each pixel is based on a patch around the pixel rather than the intensity values of the pixel. List the set of parameters of your algorithm. Discuss the pros and cons of using a patch rather than a pixel to model the pixel background value.

Answer 8

As learned in class, using a single threshold can be problematic because pixel intensities can be of a multi-modal distribution:



Algorithm Steps:

1. **1st Classification:** use a mixture of gaussians for each pixel to classify it as background or foreground as learned in class.
   Parameters: $\forall i \in [1, k]$: μ_i, σ_i (k gaussians are used for each pixel)
   Parameters Amount: $2k$ (k gaussians, 2 parameters each μ_i, σ_i)
   Output: Binary mask of classifications

2. Convolution

Let S be the patch size around p.
Apply a gaussian kernel of size S to the binary classifications mask.

Output: A mask of decimal numbers in $[0, 1]$, representing the probabilities of pixels to be classified as background / foreground **according to the surrounding patch**.
Parameters: $\mu_{kernel}, \sigma_{kernel}, S$
Parameters Amount: 3

3. 2nd Classification: Compare the mask to a threshold and classify pixels once again as background / foreground.
Parameters: th
Parameters Amount: 1

Let $Im \in R^{n \times m}$.

- $2k \cdot mn$ parameters are required to estimate the 1st classification step
- 3 Parameters are required for the convolution mask step
- 1 Parameter required for the 2nd classification step

Conclusion: $\mathbf{2k + 4}$ parameters are required for our proposed change detection algorithm.

Note: As learned in class, $I_B(p)$ can be calculated every i frames and updated every j frames. In this case, 2 more parameters are added - $\boldsymbol{2k+6}$ in total.

Using a patch to model pixel background value
Pros:

- **Spatial Context:** Classifying based on a single pixel value can suffer from noise. Relying on neighboring pixels can increase classification robustness to noise.

Cons:

- **Computational Complexity:** Additional convolution step and thresholding
- **Additional Parameters:** Additional parameters require additional parameters tuning (Define the optimal patch size S, and the parameters of the gaussian convolution $\mu_{kernel}, \sigma_{kernel}$)

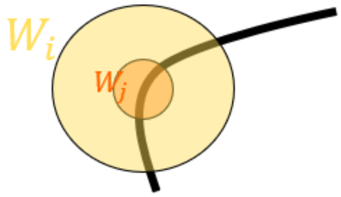
Question 9

Assume that two images of the same object is captured by two cameras with the same internal parameters, but from a different distance. Assume that in order to match feature points, the Harris corner detector is applied to the two images. Which parameter of the corner detector should be modified in order to obtain corresponding corners? Give a short explanation to your answer.

Answer 9

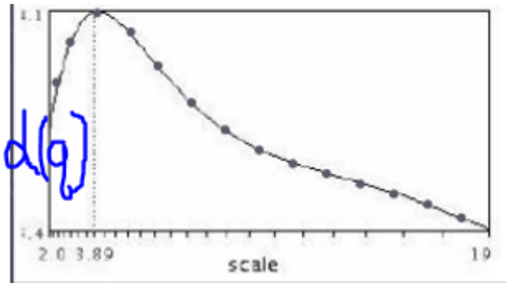
1. **Parameters:**
    - **Integration Windows $\boldsymbol{W(q)}$,**
    - **Gaussian Filter Scale $\boldsymbol{\sigma_w}$**

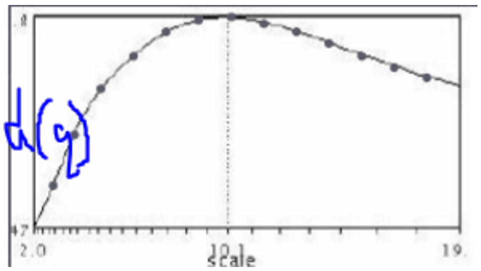
    Explanation:
    As taught in class, Harris Corner Detector is invariant to rotation and translation, but not to scale. Capturing two images of the same object from different distances will require different integration windows when searching for corners of different scales.
    For example, using a small integration window w_j results in misclassifying the corner as an edge. Using an adequate integration window on the other hand, correctly classifies the interest point as a corner.



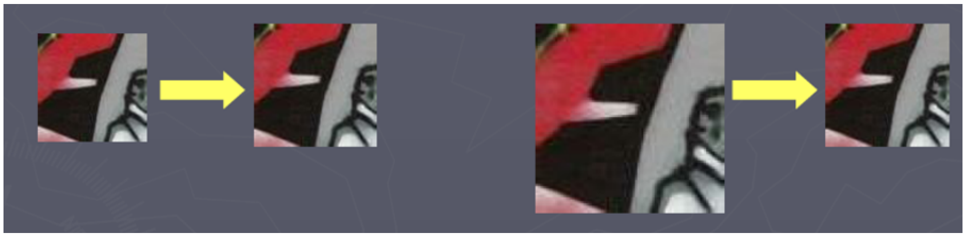
    Additionally, the filter's scale should be properly adjusted to the integration window. Using an adequate window size but a small σ_w will yield similar results to an inadequate window size.

    Formally:
    - **Corners Detection:** Detect corners using Harris' algorithm in both images, using n integration windows.
    - **Corners Strength:** Measure the corner's strength $d(q)$ for each window size w_i $(\forall i[1, n])$. If the highest $d(q)$ is small, filter it. Otherwise, choose the window size corresponding to the highest $d(q)$.





    - **Patch Size Normalization:** Normalize the corners found In different scales to a fixed size, and compare them as taught in class.



Question 10

Assume that you are given the projections of the 3D points $\{P_i\}_{i=1}^{k}$ onto two images that were captured from different locations. Let $\{p_i\}_{i=1}^{k}$ and $\{q_i\}_{i=1}^{k}$ be the sets of these projections. Moreover, you are given that pi and qi are corresponding points and $k > 50$. Suggest a method to test whether $\{P_i\}_{i=1}^{k}$ are located on a single plane. If your method requires parameters, list them and explain how they affect the results.

Answer 10

As learned in class, a homograph between two images exists either if the scene is planar or if the cameras are in the same location.

Since the images were captured from different locations:
Assumption: a homograph exists only if the scene is planar.

Proposed Method:
Step 1: H estimation
Using RANSAC to estimate H the homograph matrix:

A) Choose randomly selected points $\{p_a, p_b, p_c, p_d\}$ with their corresponding points $\{q_a, q_b, q_c, q_d\}$

B) Estimate the best 8 variables of $H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$ that conform to the constraint $q_i = H \cdot p_i = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} \mathrm{p_x} \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} \mathrm{q_x} \\ q_y \\ w \end{pmatrix}$.

C) Using the other points, compute the inliers where $\|q_i - Hp_i\| < \epsilon$

D) Loop N times and keep the largest set of inliers.

If the corresponding points indeed are on a planar surface, the estimation of H should be accurate (4 corresponding points required, >50 used for robustness).

Step 2: H, outputs comparison
For each corresponding points p_i, q_i:
Use H to estimate the corresponding point $\hat{q_i}$.
If the assumption holds, the estimated H should translate points from the first image, p_i to the corresponding points in the second image q_i, i.e.:

$$\hat{q_i} \cong q_i$$

Step 3:
Count the amount of inliers $(n = \hat{q_i} \cong q_i)$. If $n > th$, deduce that $\{P_i\}_{i=1}^{k}$ lay on a planar surface.

Parameters:
$\boldsymbol{\epsilon}$ **:** The threshold value to determine when a data point fits the RANSCA model .
N: The number of iterations for the RANSCA iteration.
$\boldsymbol{th_{distance}}$: Compare predictions with ground truth to a threshold - $\left\|\hat{q_i} - q_i\right\| < th_{distance}$
$\boldsymbol{th_{inliers}}$: Compare the number of inliers to an empirically tuned threshold. If $n > th_{inliers}$, deduce that the points lay on a planar surface.