CS4243

Computer Vision & Pattern Recognition

AY 2023/24

Lab Session 6

NUS National University of Singapore | School of Computing

Arrangement

- Part 1 – Quick Recap from the Lecture (~20 min)
- Part 2 – Lab Tutorial (~40 min)
- Break (10 min)
- Part 3 – Lab Solution (~40 min)

Lab Materials

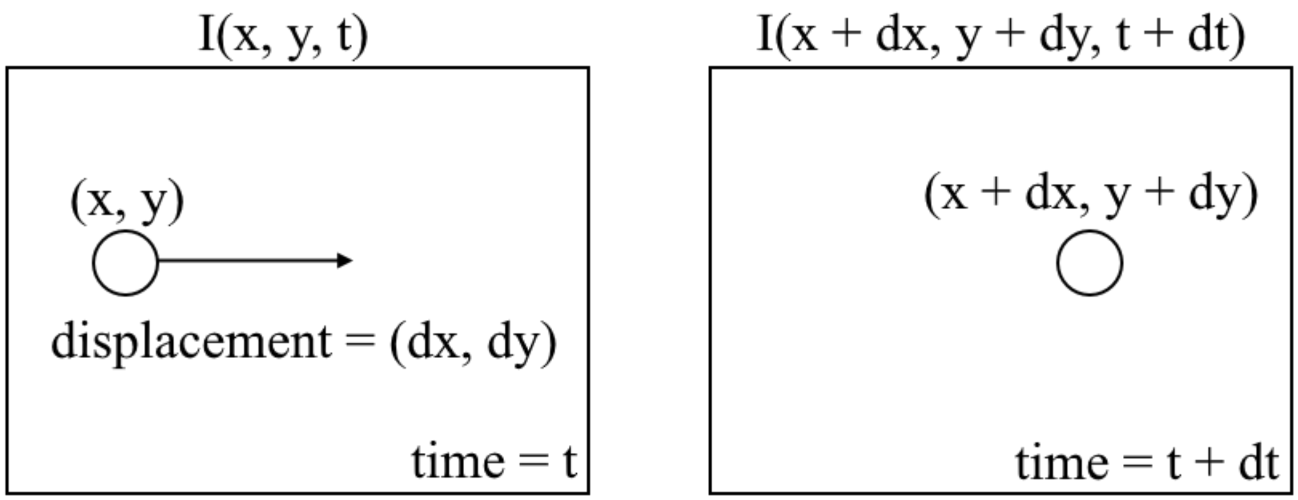
- GitHub Repo: https://github.com/ldkong1205/cs4243_lab
- Slides
- Notebook & Solution
- Other Materials (image, media, etc.)

Lesson 5

Motion Detection and Optical Flow

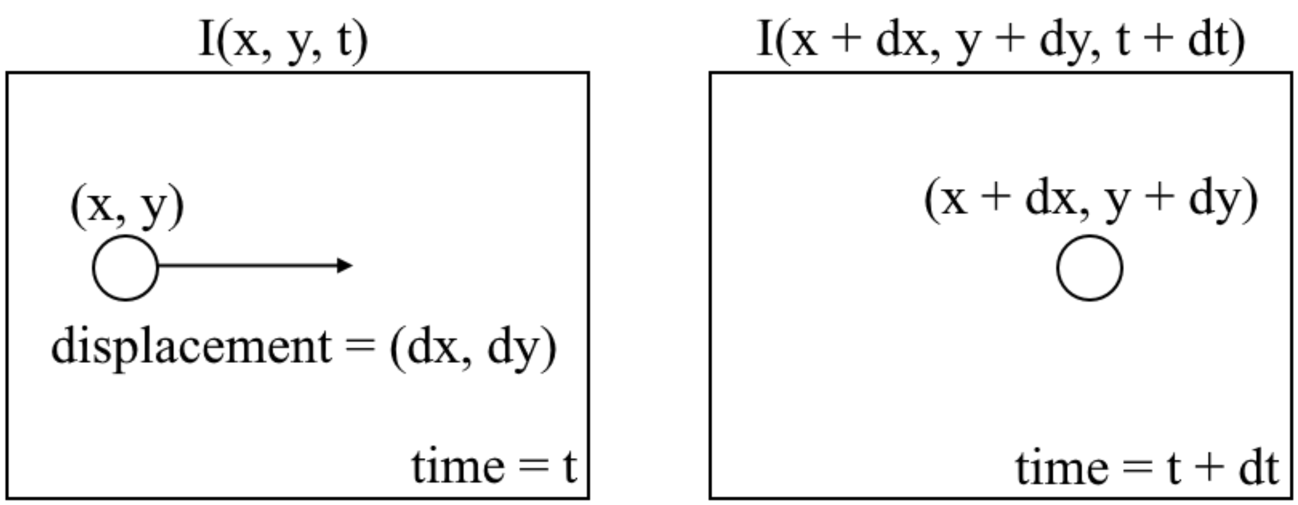
Optical Flow

Optical flow is the motion of objects between consecutive frames of sequence, caused by the relative movement between the object and camera. The problem of optical flow may be expressed as:



Optical Flow

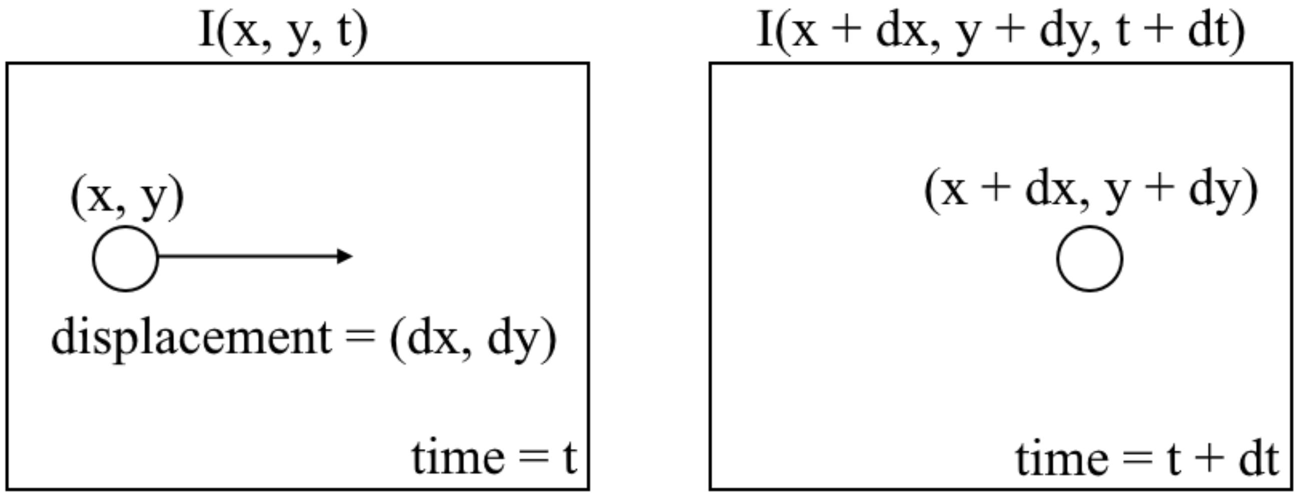
where between consecutive frames, we can express the image intensity, I, as a function of space (x, y) and time (t).



Optical Flow

In other words, if we take the first image I(x, y, t) and move its pixels by (dx, dy) over t time, we obtain the new image:

$$I(x + dx, y + dy, t + dt)$$



Optical Flow

First, we assume that pixel intensities of an object are constant between consecutive frames:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

Optical Flow

First, we assume that pixel intensities of an object are constant between consecutive frames:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

Second, we take the Taylor Series Approximation of the RHS and remove common terms:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + \ldots$$

$$\Rightarrow \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

Optical Flow

Third, we divide by dt to derive the optical flow equation:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0$$

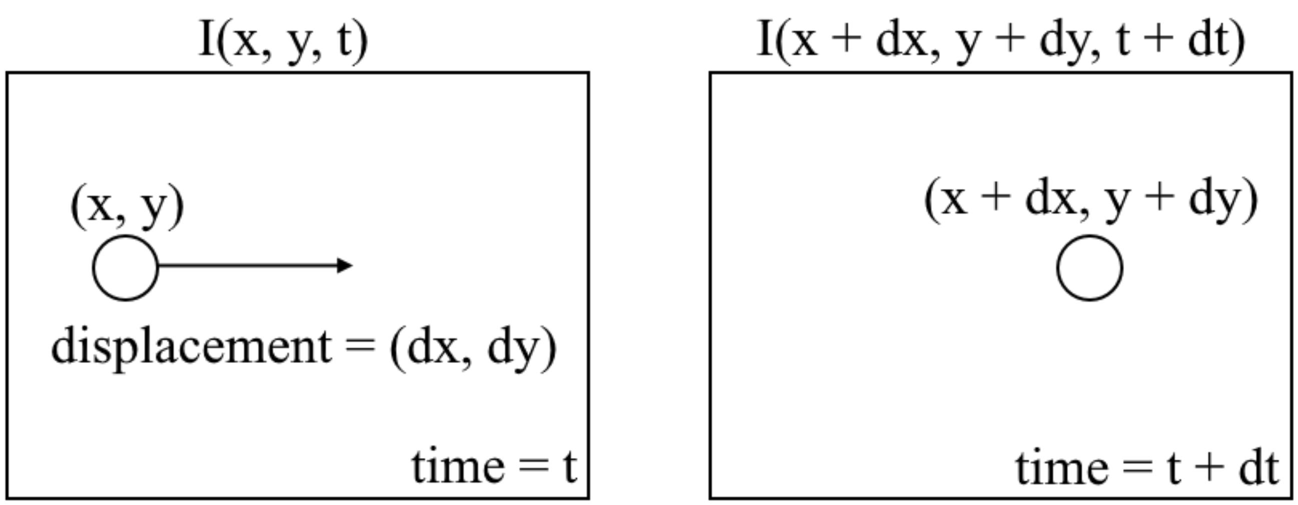
where u = dx/dt and v = dy/dt.

dI/dx, dI/dy, and dI/dt are the image gradients along the horizontal axis, the vertical axis, and time.

Optical Flow

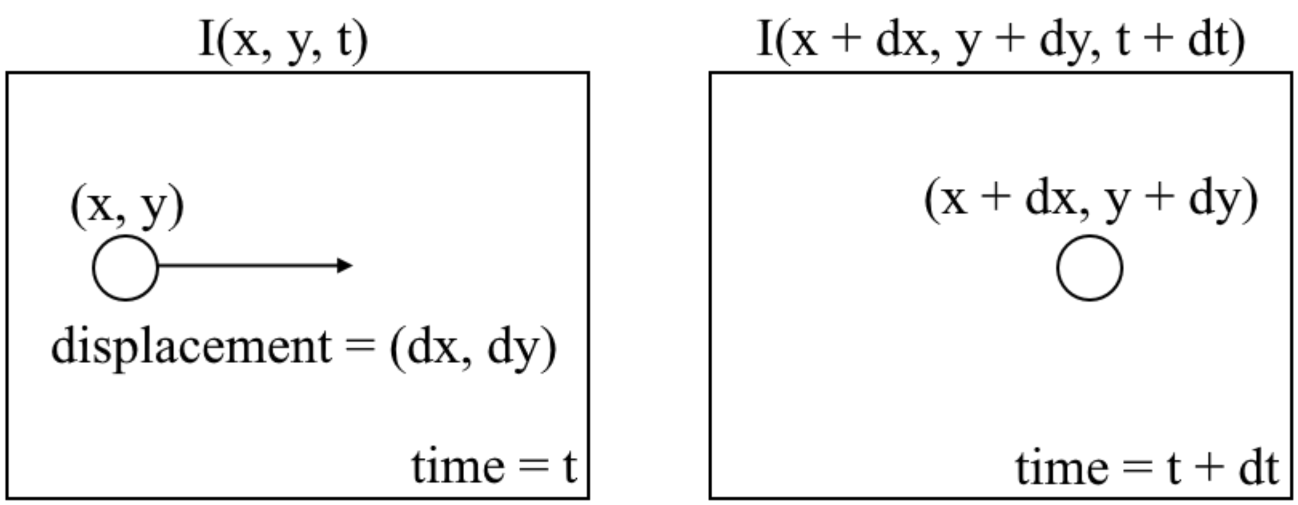
Summary:

Optical flow -> Solving u(dx/dt) and v(dy/dt) to determine movement over time.



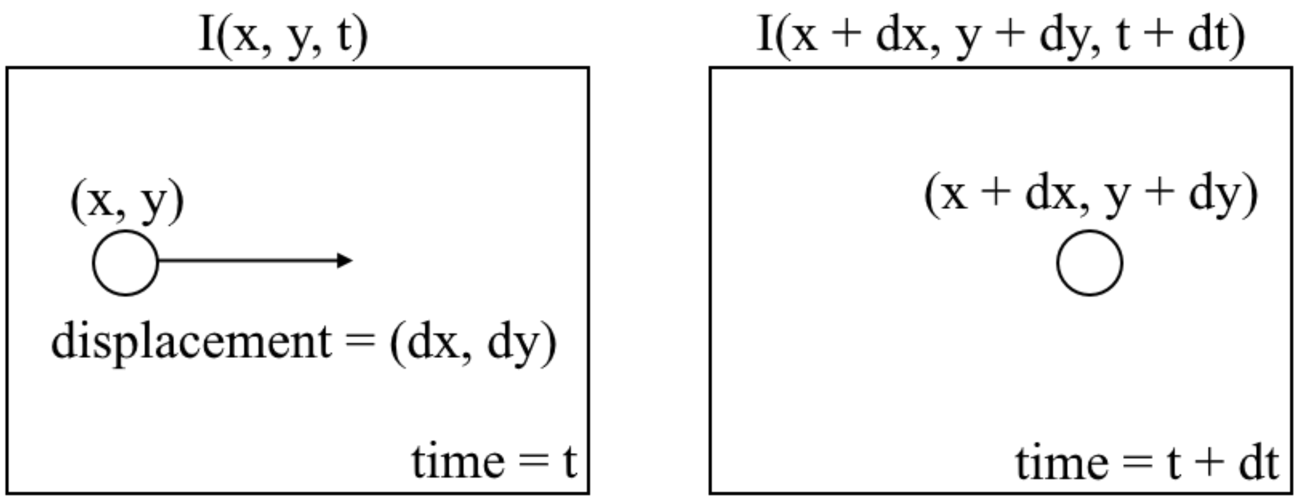
Optical Flow

You may notice that we cannot directly solve the optical flow equation for u and v, since there is only one equation for two unknown variables.



Optical Flow

In today's lab, we will implement some methods such as the Lucas-Kanade method to address this issue.



Sparse *vs.* Dense Optical Flow

Left: Sparse Optical Flow – track a few "feature" pixels.

Right: Dense Optical Flow – estimate the flow of all pixels in the image.

Tracking Specific Objects

There might be scenarios where you want to only track a specific object of interest, or one category of objects.

Lucas-Kanade: Sparse Optical Flow

Lucas and Kanade proposed an effective technique to estimate the motion of interesting features by comparing two consecutive frames in their paper:

"An Iterative Image Registration Technique with an Application to Stereo Vision," IJCAI, 1981.

Lucas-Kanade: Sparse Optical Flow

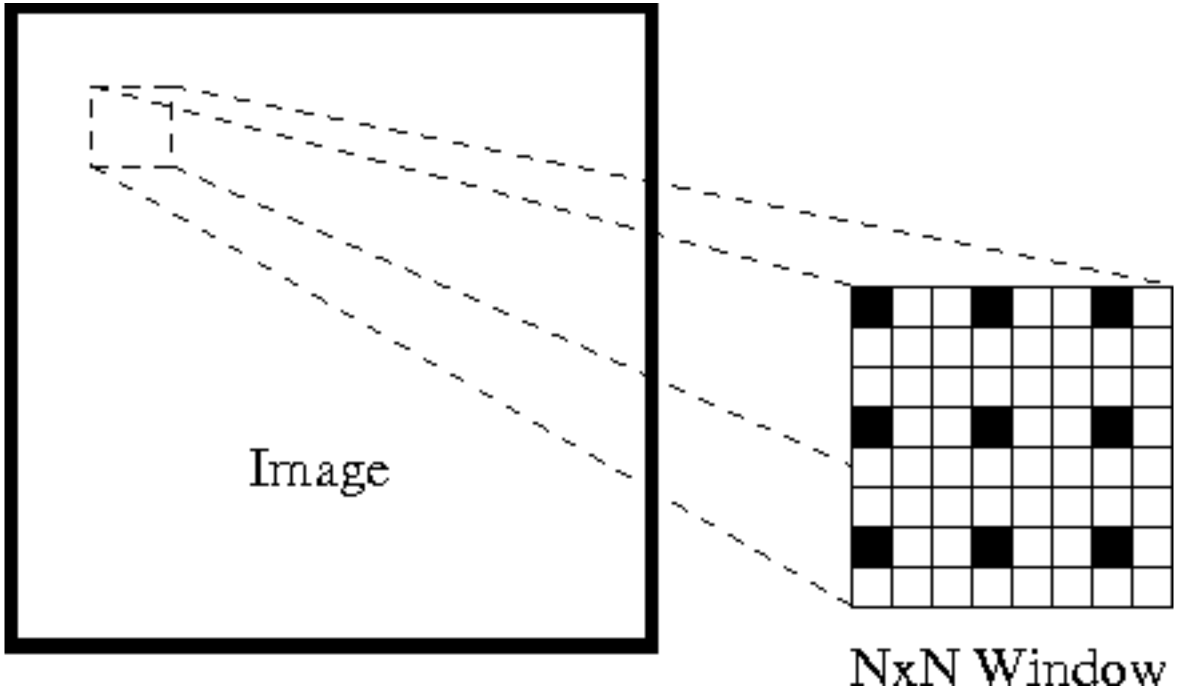
The Lucas-Kanade method works under the following assumptions:

1. Two consecutive frames are separated by a small time increment (dt) such that objects are not displaced significantly (in other words, the method work best with slow-moving objects).

2. A frame portrays a "natural" scene with textured objects exhibiting shades of gray that change smoothly.

Lucas-Kanade: Sparse Optical Flow

First, under these assumptions, we can take a small 3x3 window (neighborhood) around the features detected and assume that all nine points have the same motion.



Lucas-Kanade: Sparse Optical Flow

This can be represented as:

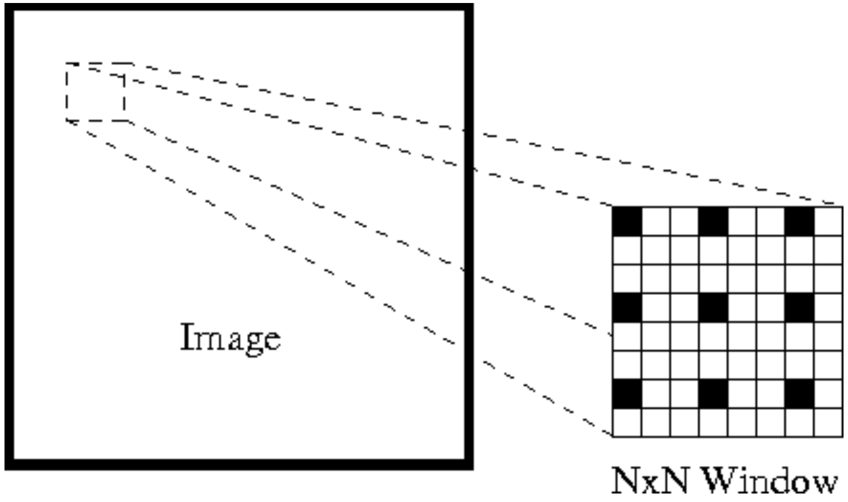
$$I_x(q_1)V_x + I_y(q_1)V_y = -I_t(q_1)$$

$$I_x(q_2)V_x + I_y(q_2)V_y = -I_t(q_2)$$

$$\vdots$$

$$I_x(q_n)V_x + I_y(q_n)V_y = -I_t(q_n)$$

(Nine-pixel intensity)

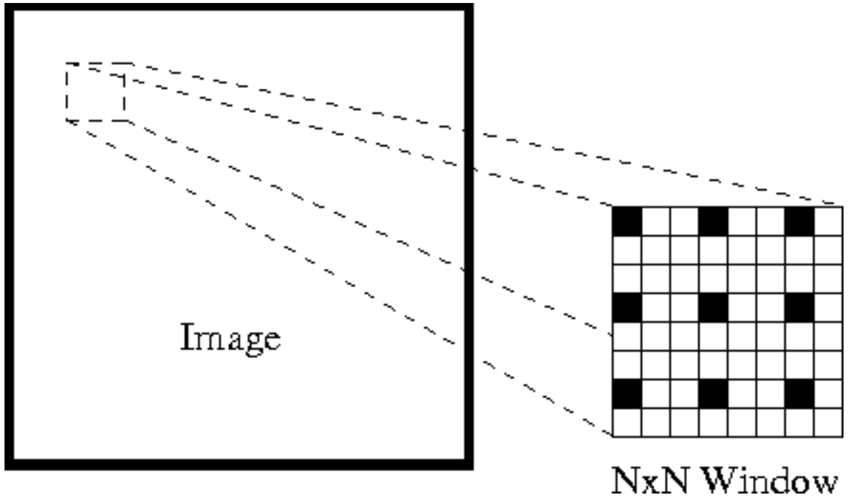


Lucas-Kanade: Sparse Optical Flow

This can be represented as:

$$I_x(q_1)V_x + I_y(q_1)V_y = -I_t(q_1)$$
$$I_x(q_2)V_x + I_y(q_2)V_y = -I_t(q_2)$$
$$\vdots$$
$$I_x(q_n)V_x + I_y(q_n)V_y = -I_t(q_n)$$

(Nine-pixel intensity)



where q1, q2, …, qn denote the pixels inside the window.

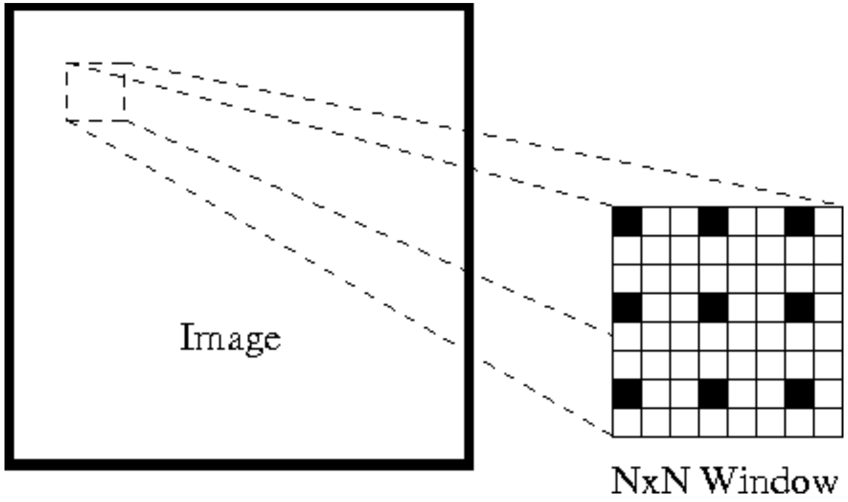
n = 9 for this 3x3 window.

Lucas-Kanade: Sparse Optical Flow

This can be represented as:

$$I_x(q_1)V_x + I_y(q_1)V_y = -I_t(q_1)$$
$$I_x(q_2)V_x + I_y(q_2)V_y = -I_t(q_2)$$
$$\vdots$$
$$I_x(q_n)V_x + I_y(q_n)V_y = -I_t(q_n)$$

(Nine-pixel intensity)



$I_x(q_i)$, $I_y(q_i)$, ..., $I_t(q_i)$ denote the partial derivatives of image I w.r.t. position (x, y) and time t, for pixel qi at the current time.

Lucas-Kanade: Sparse Optical Flow

This can be represented as:

$$
\begin{aligned}
I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1) \\
I_x(q_2)V_x + I_y(q_2)V_y &= -I_t(q_2) \\
&\vdots \\
I_x(q_n)V_x + I_y(q_n)V_y &= -I_t(q_n)
\end{aligned}
$$

Matric Form:

$$
A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix} \qquad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix} \qquad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}
$$

Lucas-Kanade: Sparse Optical Flow

Issue:

Having to solve for two unknowns Vx and Vy with nine equations, which is over-determined.

Matric Form:

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix} \qquad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix} \qquad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}$$

Lucas-Kanade: Sparse Optical Flow

Issue:

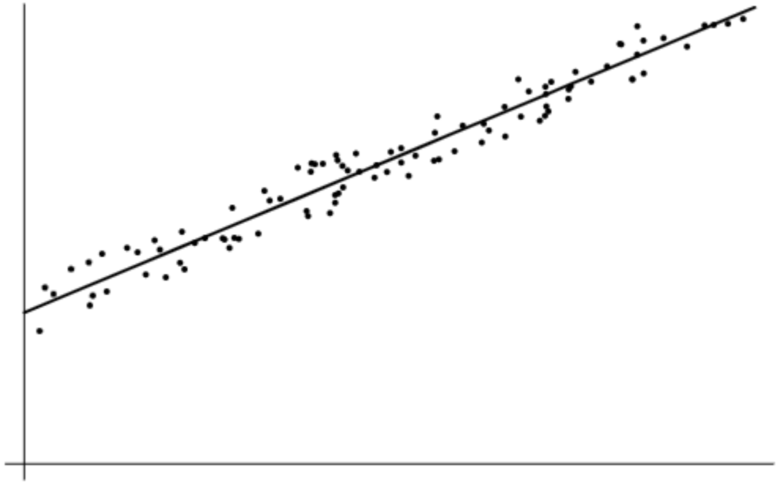
To address this over-determined issue, we apply least squares fitting to obtain the following two-equation-two-unknown problem:

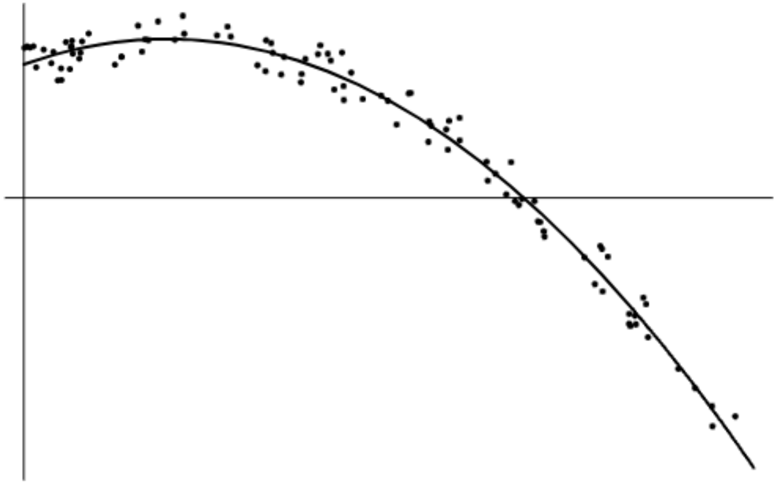
$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i) I_y(q_i) \\ \sum_i I_y(q_i) I_x(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i) I_t(q_i) \\ -\sum_i I_y(q_i) I_t(q_i) \end{bmatrix}$$

Lucas-Kanade: Sparse Optical Flow

Least Squares Fitting:

A mathematical procedure for finding the best-fitting curve to a given set of points by minimizing the sum of the squares of the offsets ("the residuals") of the points from the curve.





Lucas-Kanade: Sparse Optical Flow

Issue:

To address this over-determined issue, we apply least squares fitting to obtain the following two-equation-two-unknown problem:

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

where $V_x = u = \frac{dx}{dt}$ denotes the movement of x over time;
$V_y = v = \frac{dy}{dt}$ denotes the movement of y over time.

Lucas-Kanade: Sparse Optical Flow

Sparse optical flow of horses on a beach.

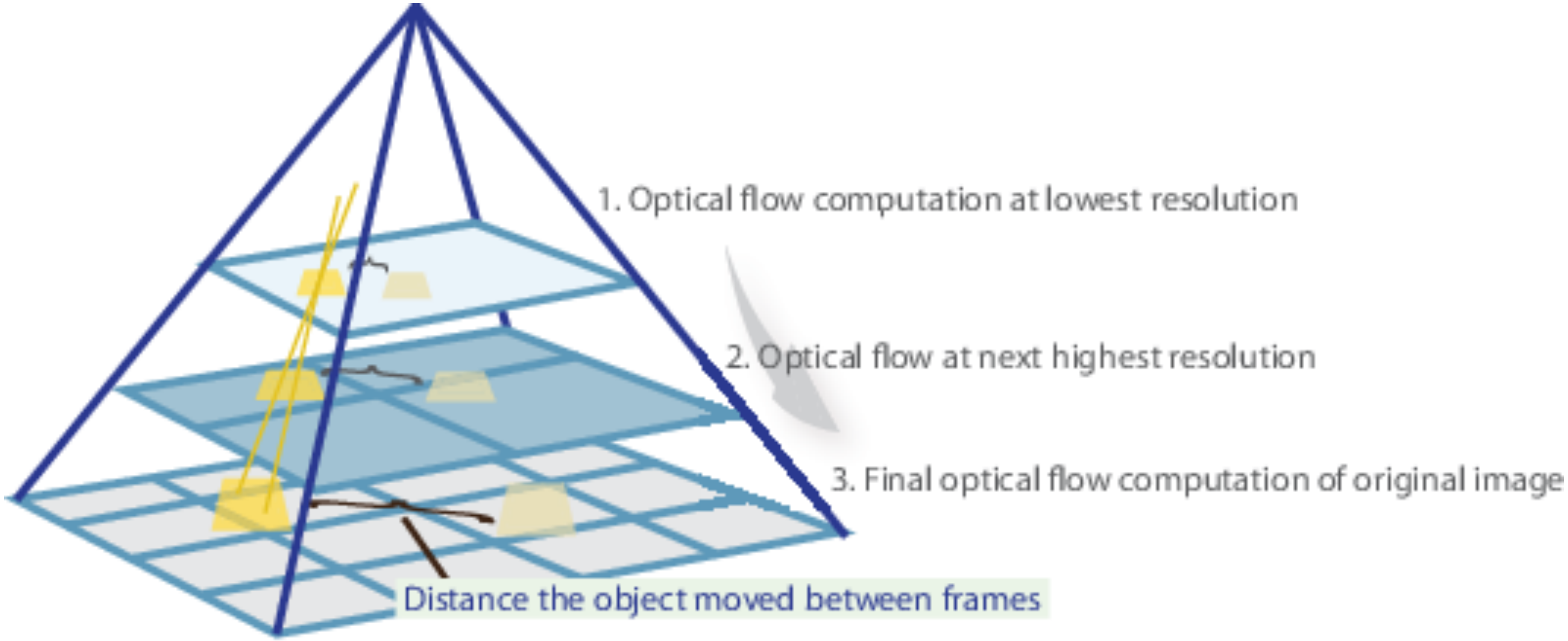
Lucas-Kanade: Sparse Optical Flow

In a nutshell, we identify some interesting features to track and iteratively compute the optical flow vectors of these points.

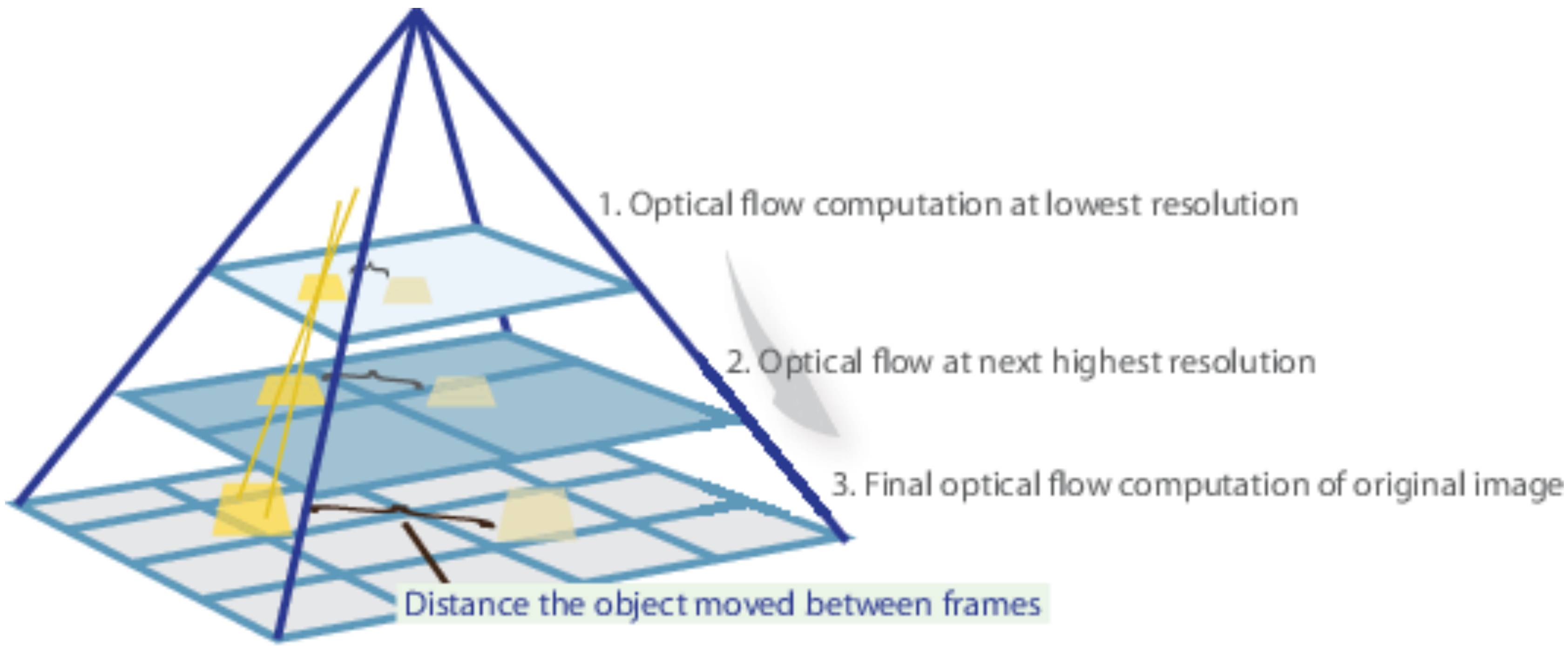
However, adopting the Lucas-Kanade method only works for small movements (from the initial assumption) and fails when there is large motion.

Therefore, the OpenCV implementation of the Lucas-Kanade method adopts pyramids.

Lucas-Kanade: Sparse Optical Flow

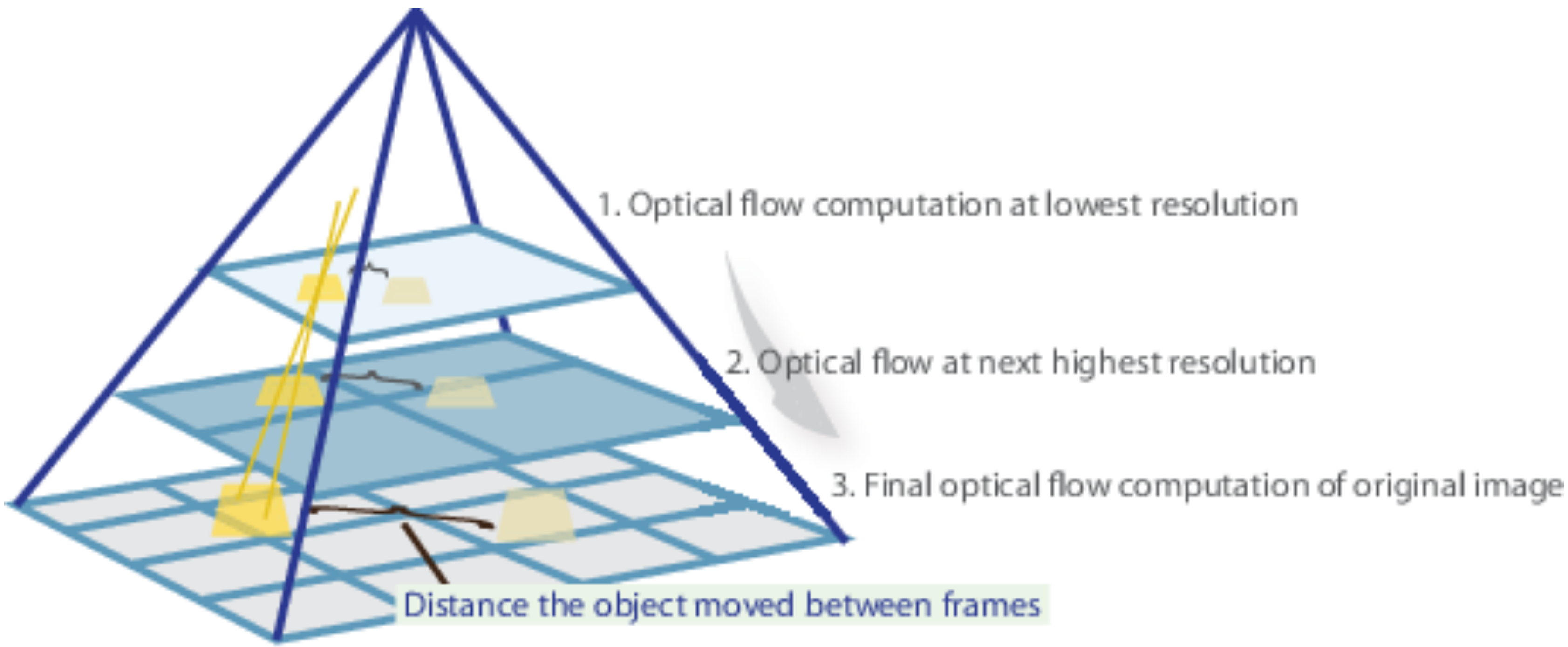


Lucas-Kanade: Sparse Optical Flow



In a high-level view, small motions are neglected as we go up the pyramid and large motions are reduced to small motions – we compute optical flow along with scale.

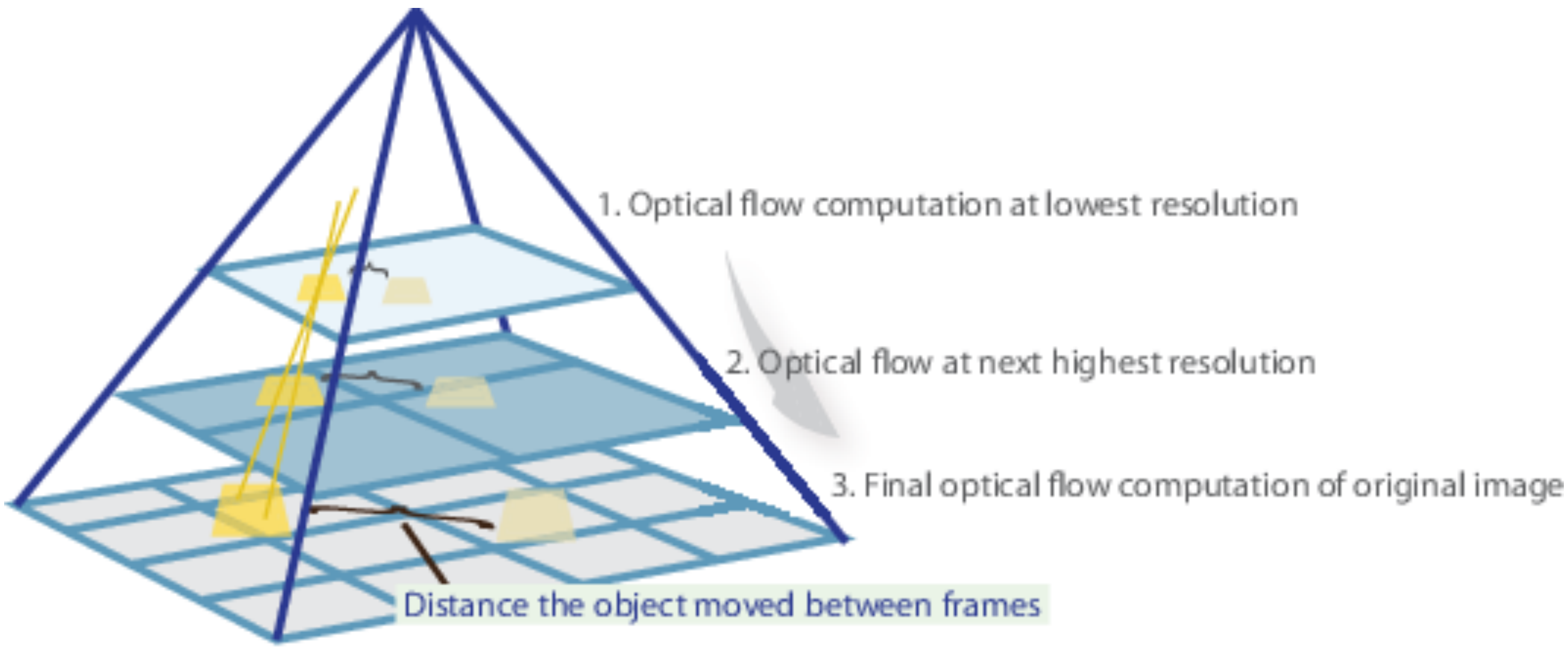
Lucas-Kanade: Sparse Optical Flow



A comprehensive mathematical explanation of OpenCV's implementation can be found in Bouguet's notes:

http://robots.stanford.edu/cs223b04/algo_tracking.pdf?ref=nanonets.com

Lucas-Kanade: Sparse Optical Flow



And OpenCV documentation of `calcOpticalFlowPyrLK()`

https://docs.opencv.org/3.0-beta/modules/video/doc/motion_analysis_and_object_tracking.html?ref=nanonets.com#calcopticalflowpyrlk

Farneback: Dense Optical Flow

Gunnar Farneback proposed an effective technique to estimate the motion of interesting features by comparing two consecutive frames in his paper:

"Two-Frame Motion Estimation Based on Polynomial Expansion," Image Analysis: 13th Scandinavian Conference, SCIA 2003.

Farneback: Dense Optical Flow

First, the method approximates the windows (similar to Lucas Kanade method of sparse optical flow estimation) of image frames by quadratic polynomials through polynomial expansion transform.

Farneback: Dense Optical Flow

First, the method approximates the windows (similar to Lucas Kanade method of sparse optical flow estimation) of image frames by quadratic polynomials through polynomial expansion transform.

Second, by observing how the polynomial transforms under translation (motion), a method to estimate displacement fields from polynomial expansion coefficients is defined.

Farneback: Dense Optical Flow

First, the method approximates the windows (similar to Lucas Kanade method of sparse optical flow estimation) of image frames by quadratic polynomials through polynomial expansion transform.

Second, by observing how the polynomial transforms under translation (motion), a method to estimate displacement fields from polynomial expansion coefficients is defined.

After a series of refinements, dense optical flow is computed.

Farneback: Dense Optical Flow

Dense optical flow of three pedestrians walking in different directions.

Farneback: Dense Optical Flow

Dense optical flow of three pedestrians walking in different directions.

OpenCV implementation:
`calcOpticalFlowFarneback()`

https://docs.opencv.org/3.0-beta/modules/video/doc/motion_analysis_and_object_tracking.html?ref=nanonets.com#calcopticalflowfarneback

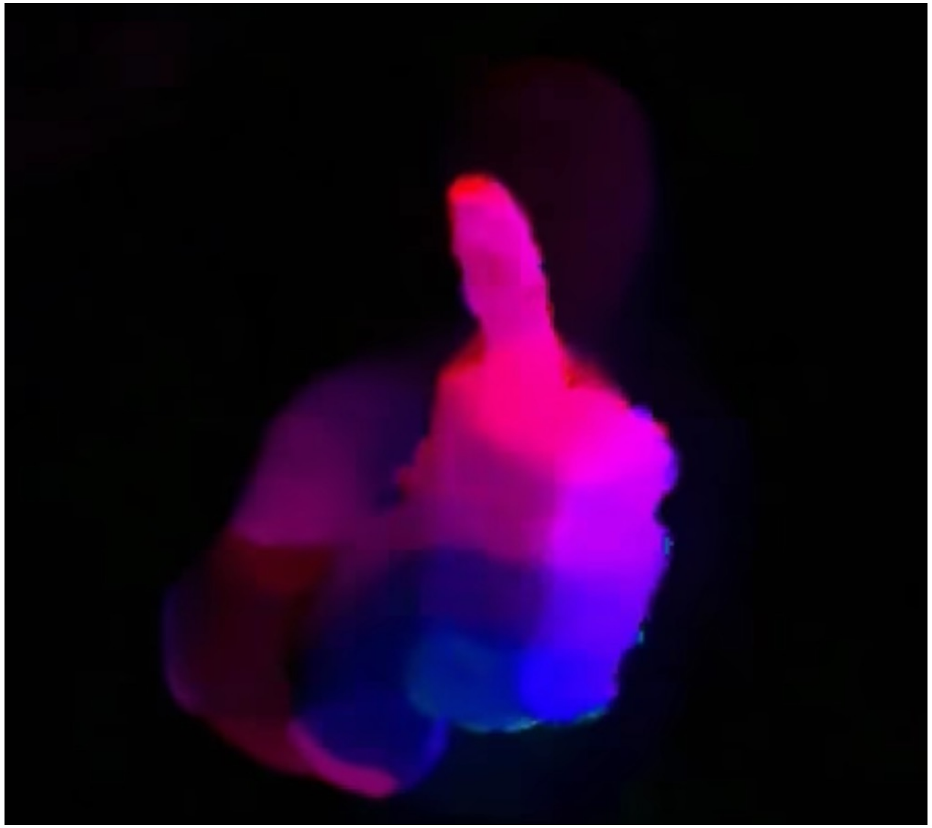
Optical Flow using Deep Learning

While the problem of optical flow has historically been an optimization problem, recent approaches by applying deep learning have shown impressive results.

Generally, such approaches take two video frames as input to output the optical flow (color-coded image), which may be expressed as:

$$(u, v) = f(I_{t-1}, I_t)$$

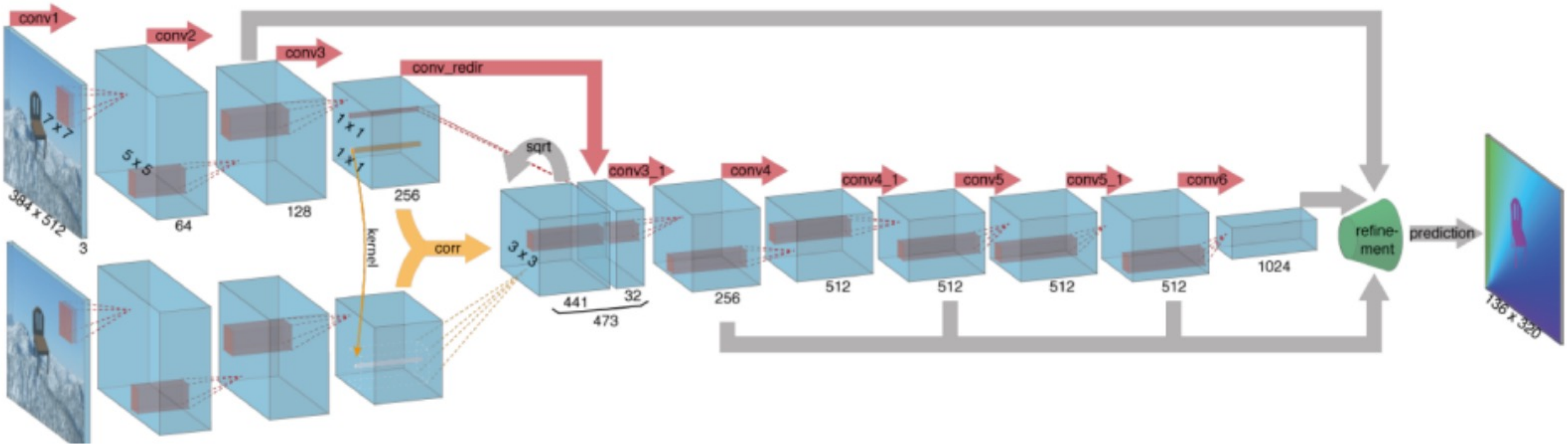
Optical Flow using Deep Learning



Output of a deep learning model: color-coded image.

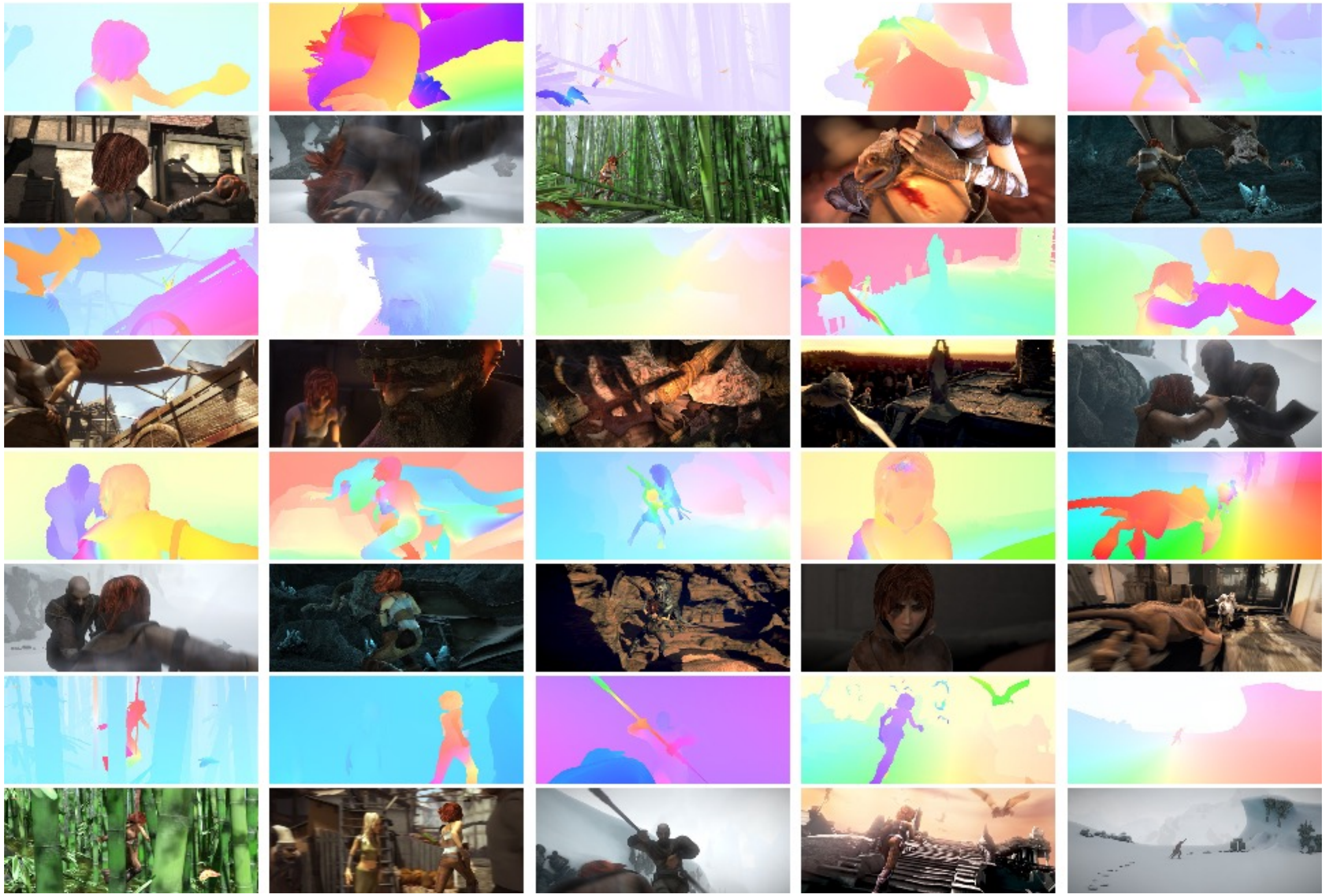
Color encodes the direction of pixel while intensity indicates their speed.

Optical Flow using Deep Learning



Architecture of FlowNetCorr, a convolutional neural network for end-to-end learning of optical flow.

Optical Flow using Deep Learning



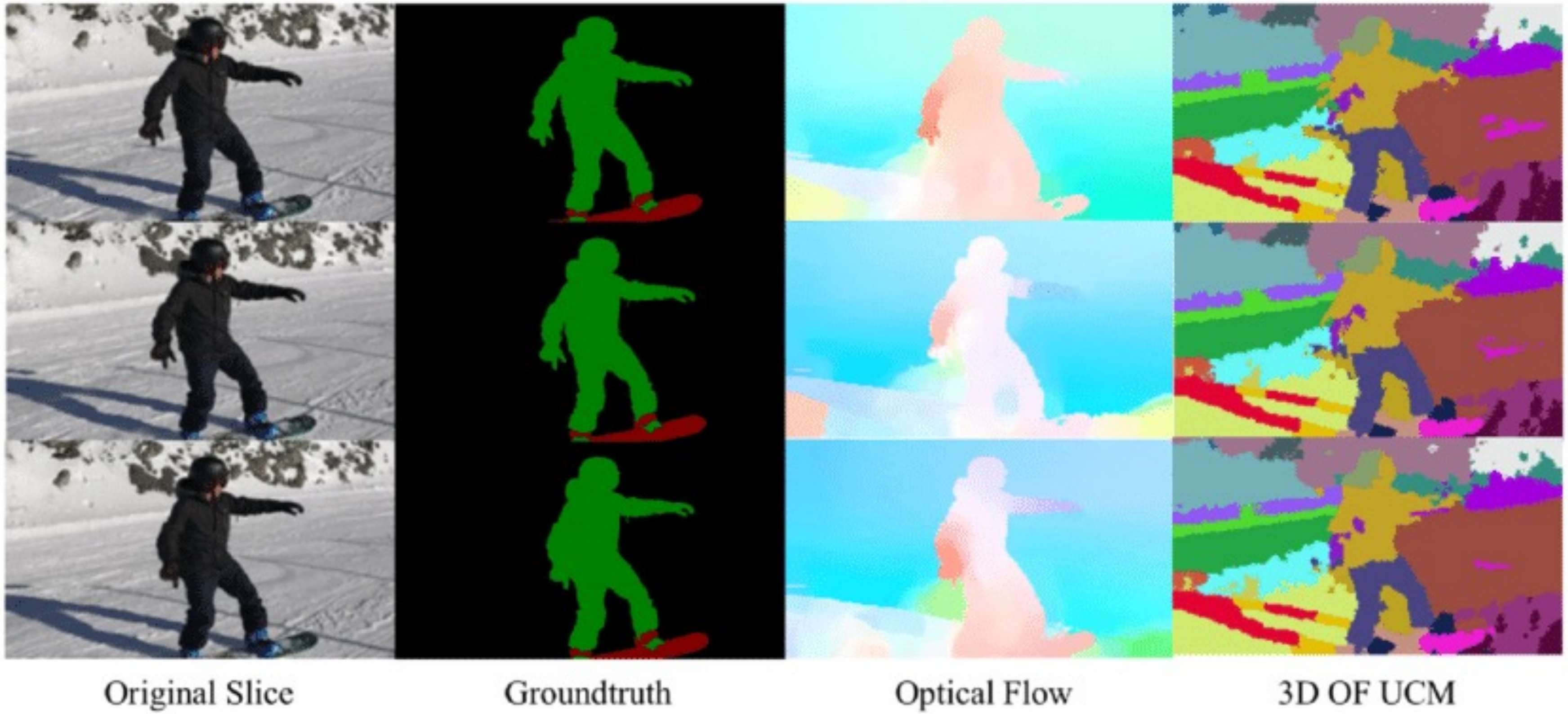
Synthetically generated data for training Optical Flow Models – the MPI-Sintel dataset.

Optical Flow using Deep Learning

Synthetically generated data for training Optical Flow Models – the Flying Chairs dataset.

NUS National University of Singapore | School of Computing

Application: Semantic Segmentation

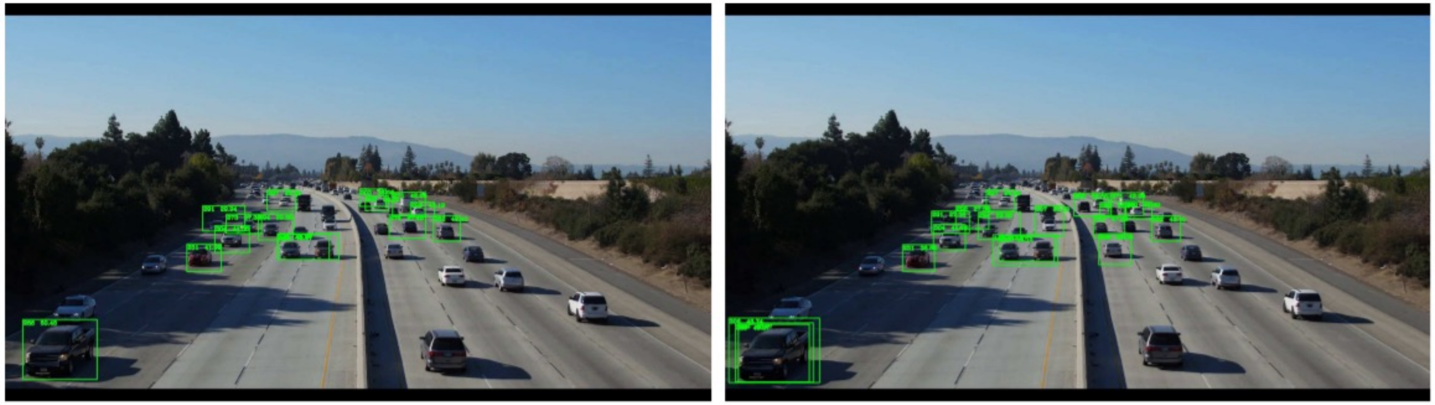


Semantic segmentation generated from optical flow.

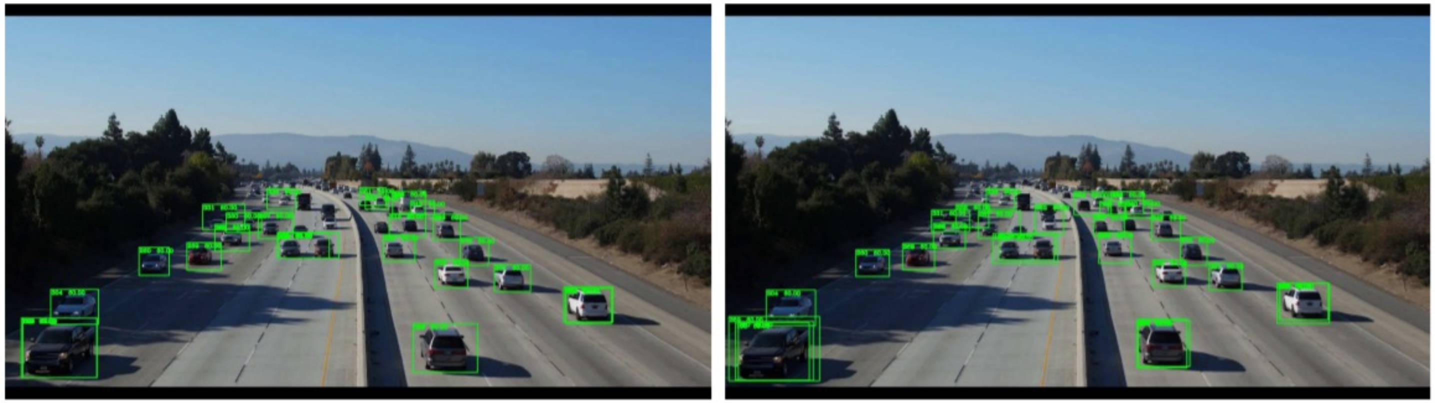
Application: Object Detection & Tracking

Real-time tracking of vehicles with optical flow.

Application: Object Detection & Tracking



(a) Predicted Speed Model



(b) Constant Speed Model

Optical flow can be used to predict vehicle speeds.

Lab Session 6

Optical Flow

NUS National University of Singapore | School of Computing