INTRODUCTION MOTION

We will focus less on images and more on pattern recognition time and space.

CV method

1- Tracking object
2- Protecting their movement

Create a filter that tracks a robot as it moves sense its environment over time.

This require having a **model of motion** and **mathematical way** to represent uncertainly in that motion.

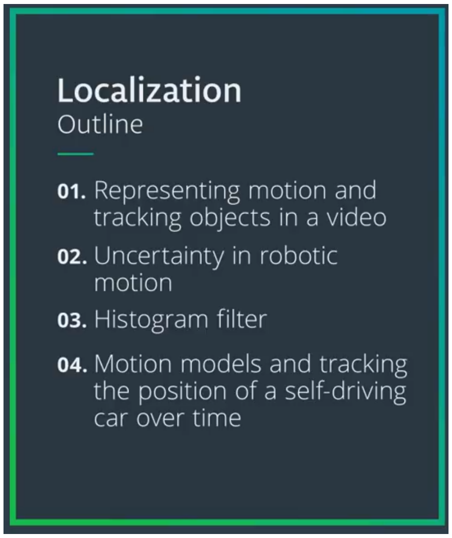
Finding out where a moving object is located with some certainly at any given time is called localization.

Localizations techniques are used extensively in computer graphics applications that track a certain objects and people over time, and using in autonomous vehicles like in self-driving cars that need to know where they are in the world so that they can safely navigate.

We'll start off by talking about representing motion and tracking objects in a video. Then, we'll move onto uncertainty and robotic motion and learn a simple localization technique, the histogram filter, which can account for this uncertainty.

Then, you'll learn about motion models and tracking the position of a self-driving car over time.

you'll have all the skills you need to code an implementation of SLAM, which stands for Simultaneous Localization and Mapping. Slam is a technique that allows autonomous vehicles to build up a model of the world while locating itself within that world.



INTRODUCTION MOTION

Motion : static image and processing them to identify objects and interesting features, and the exact same processing techniques can be used on a video stream. This is because a video stream is just made up of sequences of image frames.

<u>One thing we haven`t talked about that`s unique to these sequence of image frames is the idea of motion.</u>

<u>When watching this video you can see or envision my hands move and to create an intelligent computer vision system, we also want to give a computers a way to understand motion.</u>

<u>This is useful in a number of applications.</u>

> *"Knowledge about motion is used to isolate moving pedestrians from a still background."*

It is used in ***<u>intelligent navigation system</u>*** in ***<u>movement prediction model</u>***, and in distinguishing behaviors like running versus walking in a given video.

- One way to *<u>track objects</u>* over time and *<u>detect motion</u>* is by extracting certain features and observing how they change from one frame to the next.

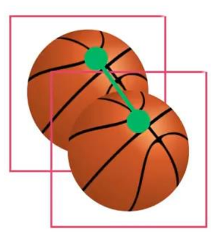
Optical flow

Optical flow: it is used in <u>many tracking motion</u> and <u>analysis applications</u>.

- ***It works by assuming two things about image frame.***
    - ***The pixel intensities of an object do not change between consecutive.***

Two consecutive frames

    - ***Neighboring pixels have similar motion (it then looks at interesting points, say concern or particularly <u>bright pixels</u>, and track them from one frame to the next.)***



Two consecutive frames

- ***Tracking a point or a set of points provides information about how fast them point or object is moving and in what direction. (<u>This is also allows you to predict where an object will move next.</u>) So you can use optical flow to do things like***

1- *Hand gesture recognition or*
2- *it to track a certain object like person or vehicles*

Motion recognition can be used to distinguish behaviors like <u>running</u> versus <u>walking</u> and in safely applications <u>by predicting the motion</u> of things and performing <u>obstacles avoidance</u> like in the case of self-driving cars.

It`s even used in eye tracking for virtual reality games and advertising.

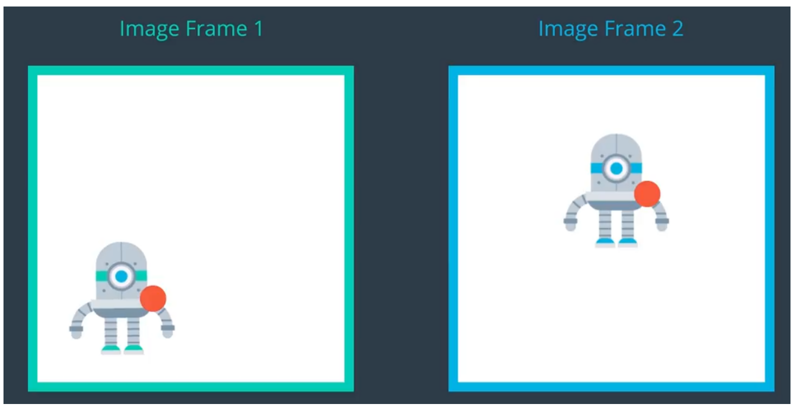
So, in many applications tracking and motion can add some very valuable information.

How exactly work ?

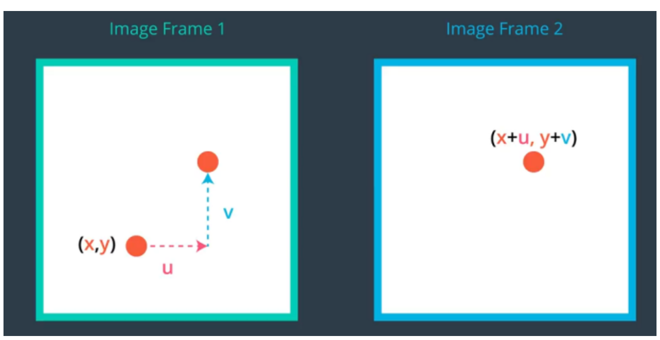
1- ***we know that it tries to <u>track points</u> from one image frame to another based on the intensity levels of points in each image.***

Example

1- ***We have two image frames from a video, and for one point on object in image one,***
   a. ***We want to find out where it is in image two. (once we do that, we calculate a motion vector that describes the velocity of this point from the first frame to the next.)***

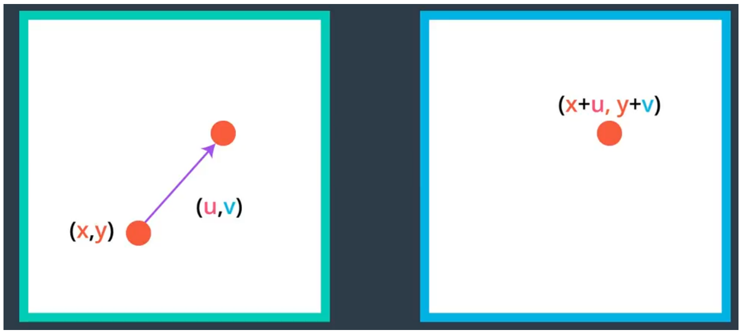


The math work like this, a point in our first image(x , y) will move some amount from this frame to the next. It will have moved some distance u horizontally and some distance v vertically so in the second image frame, that point will be at the coordinate x + u and y + v



INTRODUCTION MOTION

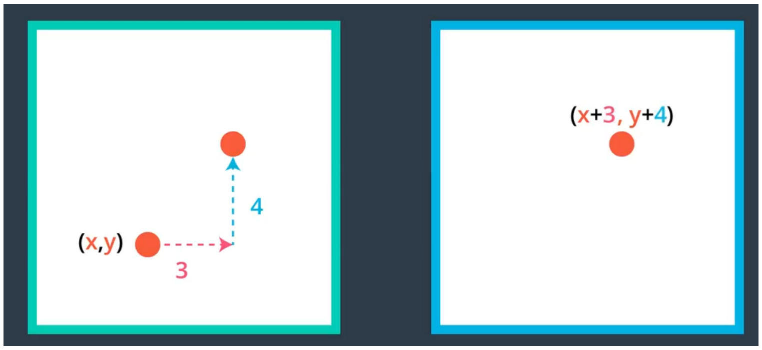
This is motion can be described by a motion vector (u , v).



The vector is a quantity that has a magnitude and a direction.

<u>Example</u>

- ***<u>Let`s see that a point has moved three pixels to the right and four pixel up, then our point(x ,y) in the first frame will be at x + 3 and y + 4 in the second frame.</u>***

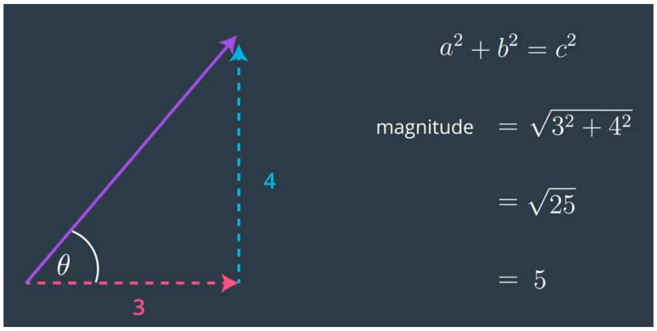


<u>The motion vector will be (3,4).</u>

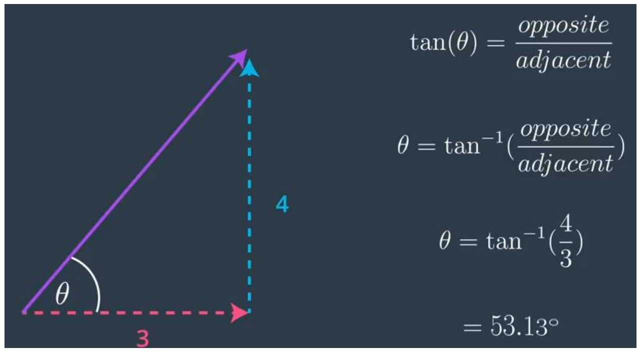
The magnitude of any vector x , y space can be found by the <u>Pythagorean</u> theorem for sides of a triangle.

1- *A-squared plus b-squared equals c-squared. Pow(a,2) + pow(b,2) = pow(c,2). So, in this case, the magnitude of the vector equals the square root of 3 squared + 4 squared, that`s the squared root of 25, which is five*

The orientation of this vector can be found using <u>trigonometry</u>.



Where the angle equals the inverse tangent of four over three which is about 53.13 degree.



Knowing the magnitude and direction of a <u>moving point</u> or a <u>set of moving point</u>, is all you really need to know to track an object.

How optical flow estimates motion vectors like these?

Optical flow assume the points in one image frame have the same intensity pixel values as those points in the next frame.

That is optical flow assumes that the color of a surface will stay the same over time.

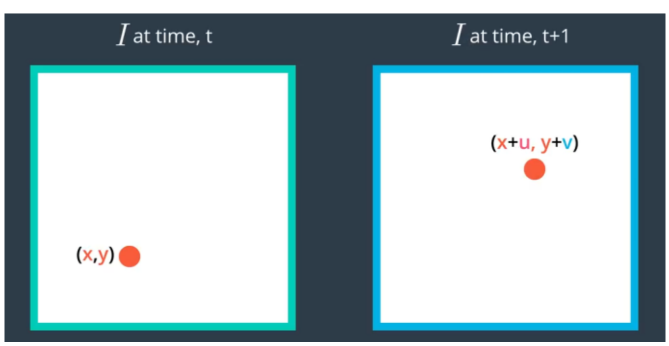
In practice , this is not a perfect assumption but it`s also most of the time.

So, in this image which I will call I_1 and I_2, the intensity at the point x, y in this first image is the same as the intensity in image 2 at the point x + u , y + v. So far, we have treated as two separate images in x and y space, <u>but we also that they are related in time</u>.

$$I_1(x, y) = I_2(x + u, y + v)$$

<u>How do you think we can mathematically represent that images one comes right before image two?</u>

- *To relate image frames in space and time, we can think about these image frames in another way.*
  - *The first image is just the 2d pattern of intensity that happens at time t, and the second image is the intensity pattern that happens at time t + one, one-time step later.*



In this way, we can think of a series of image frames I as a ***3D volume of images with x and y coordinates, pixel value at each point, and depth dimension*** **of time**.

We can write this intensity equation as a function of x and y and t. ***The equation is known as the brightness constancy assumption.***

$$I(x, y, t) = I(x + u, y + v, t + 1)$$

This function can be broken down into something called a ***Taylor series expansion****, which represents this* ***intensity as a summation of terms****.*

In this case, the terms ***I calculated as the derivatives of the intensity with respect to x, y and t****.*

Brightness Constancy Assumption

$$I(x, y, t) = I(x + u, y + v, t + 1)$$

Taylor Series Expansion

$$I(x, y, t) = \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}1 + I(x, y, t)$$

We can simplify this expansion. And the result is an equation that relates the ***motion vector quantities*** *u and v to the change in image intensity in space and in time, which are measurable changes. This is the foundation of how optical flow estimates the motion vectors for a set of feature point in a video.*

Brightness Constancy Assumption

$$I(x, y, t) = I(x + u, y + v, t + 1)$$

Taylor Series Expansion

$$0 = \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}1$$

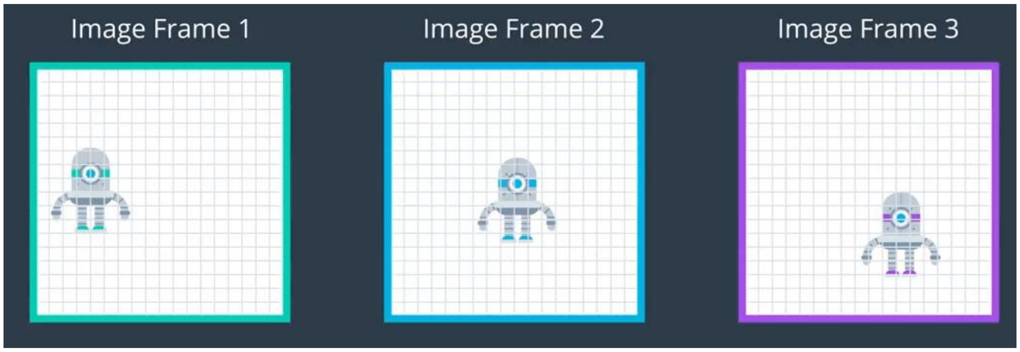
$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v = -\frac{\partial I}{\partial t}$$

INTRODUCTION MOTION

what does it look like when you apply optical flow not just to a point, but a set of points in a video?

- *Take a small image of robot and then down into the right and the next image frames.*

The goal of optical flow Is, for each image frame, to compute approximate motion vectors based on how the image intensity has change over time. (the pattern of dark and light pixel has changed over time).



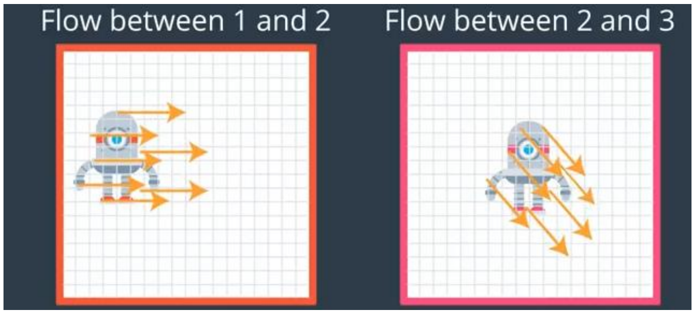
1- *The first step is to find matching feature points in between two images using a method by **cog** or **corner detection** that looks for **matching pattern of intensity**.*

Corner detection *is an approach used within computer vision systems to extract certain kinds of features and infer the contents of an image. Corner detection is frequently used in motion detection, image registration, video tracking, image mosaicing, panorama stitching, 3D reconstruction and object recognition. Corner detection overlaps with the topic of **interest point detection**.*

In this case, we will directly detect some endpoints and maybe the sensors on the robot.

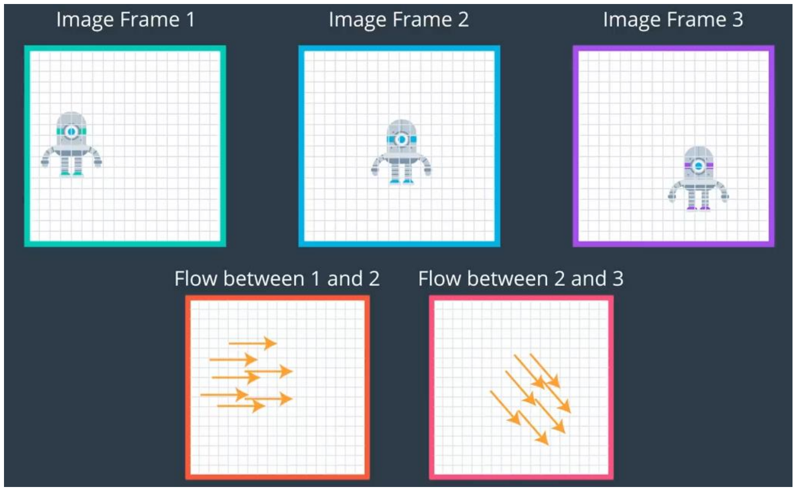
*Then optical flow calculates a motion vector, UV for each key point in the first image frame that points to where that **key point can be found in the next image**.*

*These **motion vectors** are to the right between frames one and two, and then down into the right between frames two and three.*



INTRODUCTION MOTION

*This is what **optical flow** will look like for **a set of points** and **you can calculate this flow frame by frame** until you build up the **path of an object over time**.*



*You can imagine using this **data to measure the velocity** of that object. You can also apply this technique to every pixel in an image to create a field of motion vectors.*

*Optical flow is used in a variety of applications from slow motion graphics to autonomous vehicle navigation, and it will be especially useful to keep these motion vector in mind as we approach the task of **locating a robot** as it moves through an environment.*