REAL TIME VIDEO SENSING

MODELLING AND PROCESSING

Dr TIAN Jing

tianjing@nus.edu.sg

Module objective

Module: Video sensing

Knowledge and understanding

- Understand the fundamentals of video modelling, motion feature extraction and video analytics.

Key skills

- Design, build, implement and evaluate various motion feature representation methods for real-world application
- Apply motion feature representation methods using Python and OpenCV.

Module reference

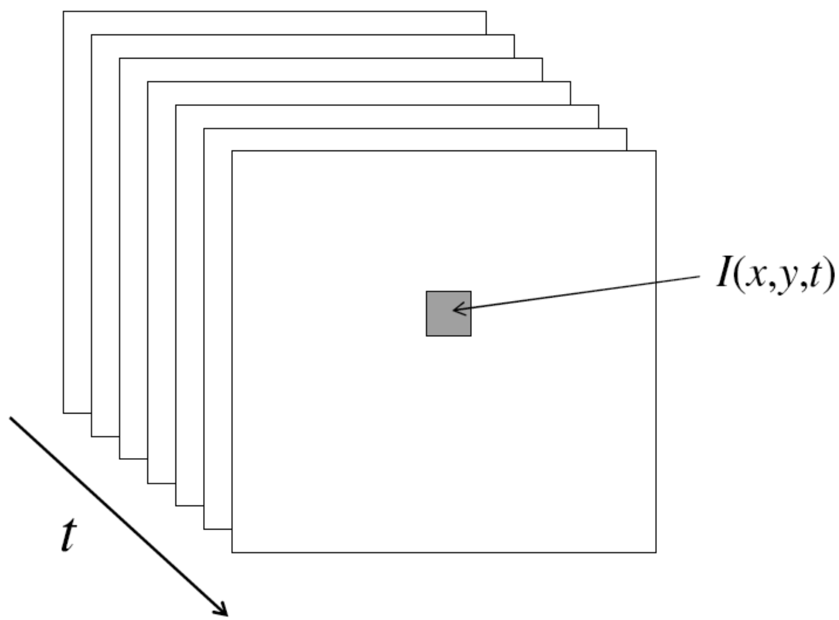
- [Intermediate] CS 4476 Introduction to Computer Vision, https://samyak-268.github.io/F18CS4476/
- [Exclusively for object tracking] Visual tracking course, Winter School 2017, https://cw.fel.cvut.cz/old/courses/ucuws17/start
- [Exclusively for object tracking] Vision-based tracking, http://www.cse.psu.edu/~rtc12/CSE598C/
- [Advanced] Computer Vision III: Detection, Segmentation and Tracking (CV3DST) (IN2375), https://dvl.in.tum.de/teaching/cv3dst-ss20/

Topics

- Fundamentals of video data modelling, and motion feature representation methods
- Object tracking in the video
- Workshop: Build motion feature extraction and object tracking in the video

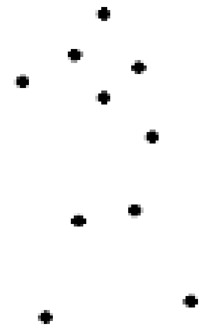
From image to video

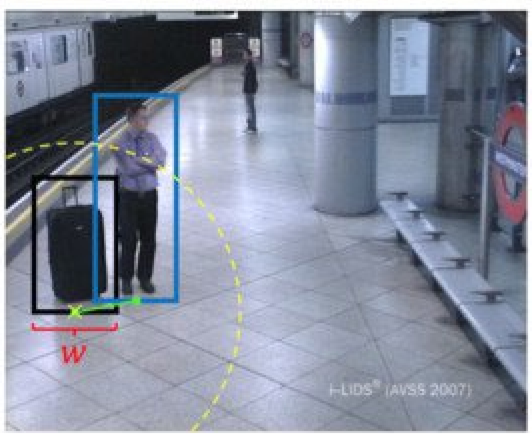
- A video is a sequence of frames captured over time. Data is a function of space (x, y) and time (t).
- Frame resolution: Dimension of each frame.
- Frame rate: The number of frames or images that are projected or displayed per second, usually measured in *frames per second* (fps).

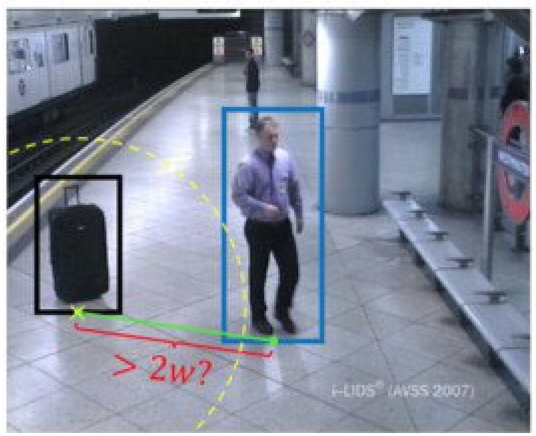


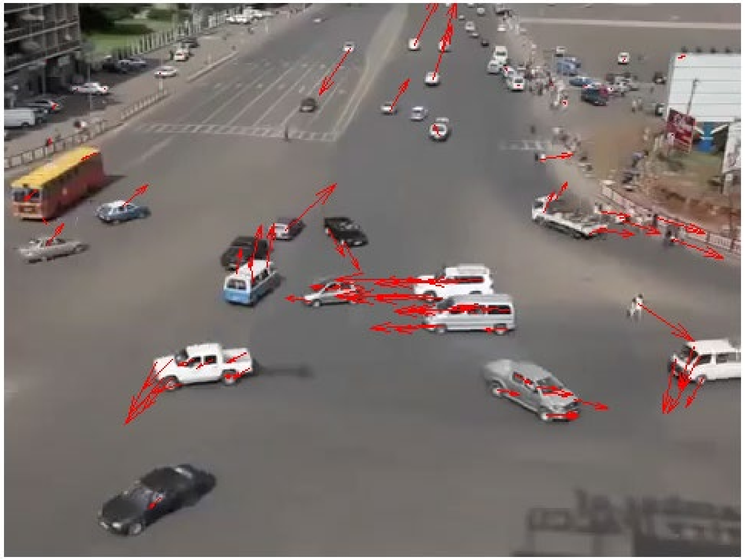
Why do we need video (compared with static image)

- See object motion
- See scene change over the time
- Track object trajectory
- Recognize event









Reference: G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics, Vol. 14, pp. 201-211, Abandoned Object Detection in Video-Surveillance: Survey and Comparison, https://www.mdpi.com/1424-8220/18/12/4290 1973.*

Motion scenarios

Static camera, moving scene

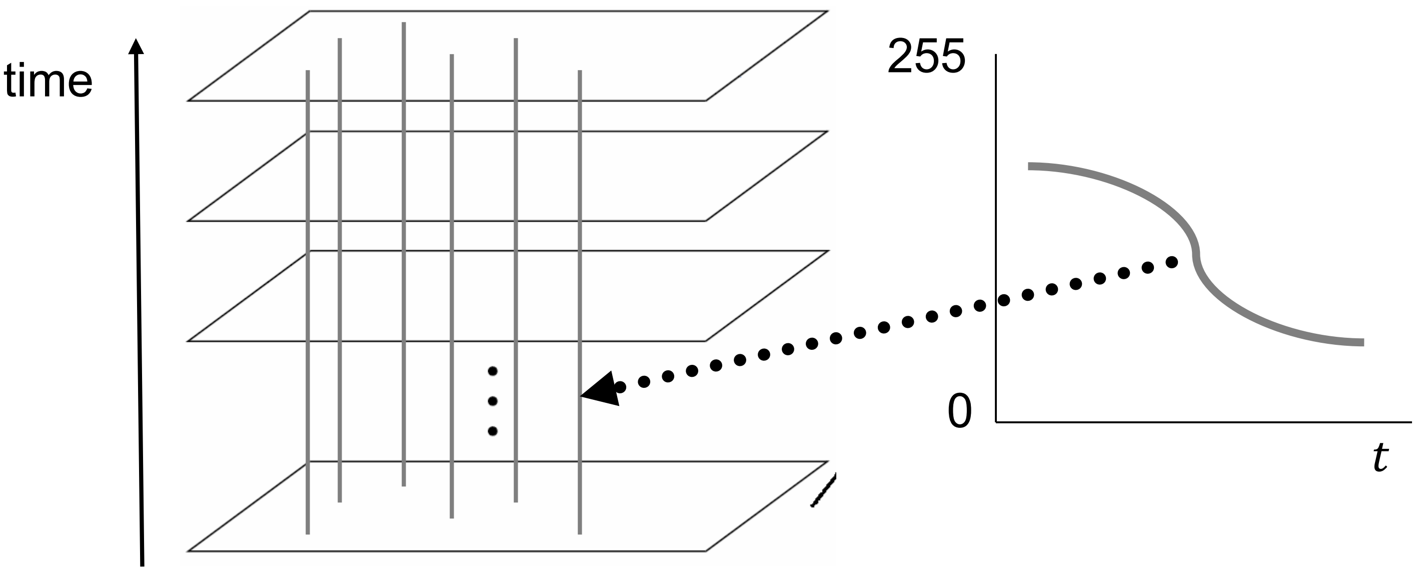
Moving camera, static scene

Moving camera, moving scene

Static camera, moving scene, moving light

Background modelling (1)

- Given a video sequence, we want to identify the foreground objects.
- We look at video data as a spatial-temporal volume



Background modelling (2)

- Estimate the background for time t (e.g., a moving average frame)
- Subtract the estimated background from the input frame
- Apply a threshold Th to the absolute different to get the foreground mask

$$\boxed{|I(x,y,t) - B(x,y,t)| \geq Th}$$

Image at time $t, I(x,y,t)$

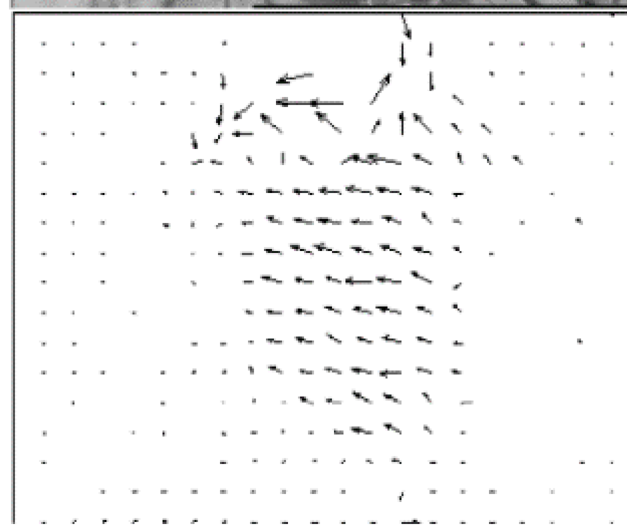
Background at time $t, B(x,y,t)$

Small threshold Th

Large threshold Th

- Advantages: Easy to implement and use!
- Disadvantages: Accuracy of frame differencing depends on object speed and frame rate. How to set threshold?

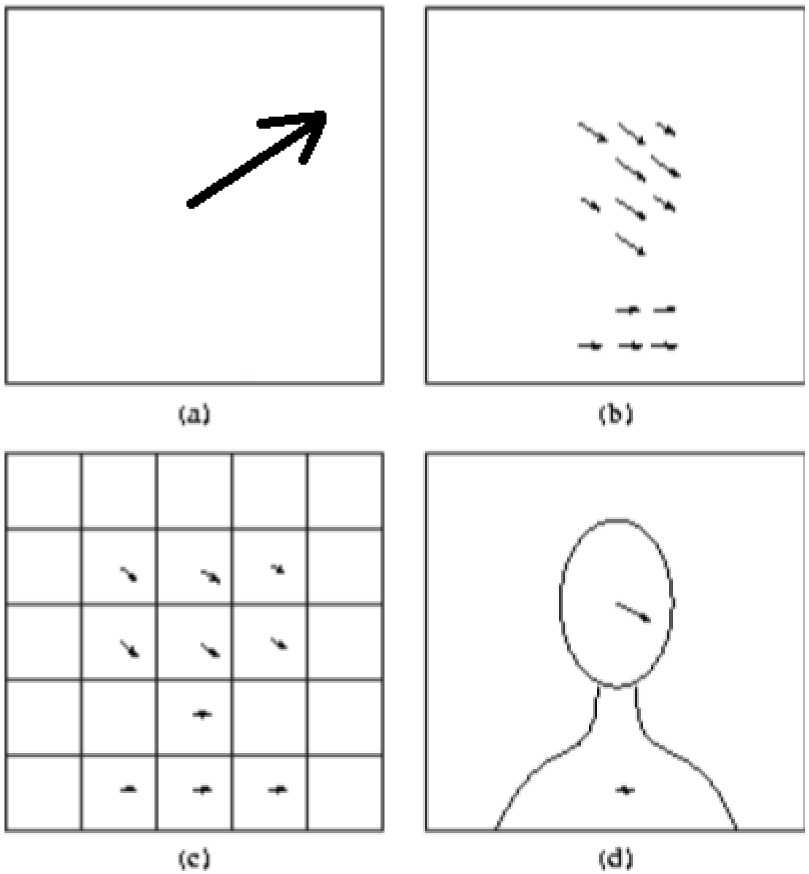
Motion representation: Motion vector



The motion vector describes the 2D displacement at the pixel location between the reference image (right) and the other target image (left).

Motion representation

Frame-based: Entire motion field is represented by a few global parameters, such as translation.



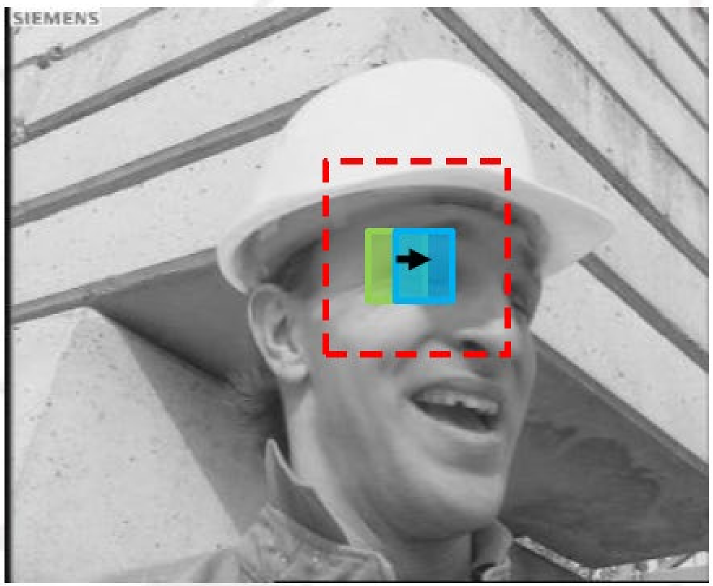
Pixel-based: One motion vector at each pixel.

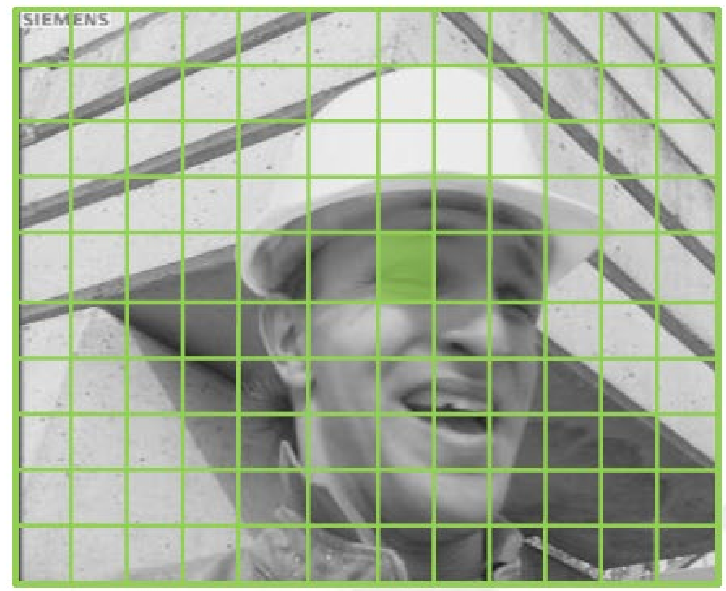
Block-based: Entire frame is divided into blocks, each block is represented by a constant motion vector.

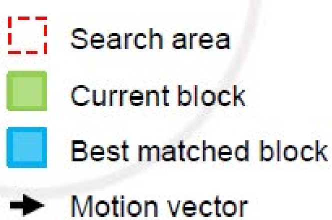
Region-based: Entire frame is divided into regions, each region is represented by an object with consistent motion.

Block-based motion feature (1)

- Idea: Assume all pixels in a block undergo a coherent motion, and search for the motion parameters for each block independently.







	(row coordinate, column coordinate)	Motion vector
Block in current frame	$(5,7)$	Either way is okay. • Current – previous: $(-3,1)$ • previous – current: $(3,-1)$
Best matched block in previous frame	$(8,6)$	

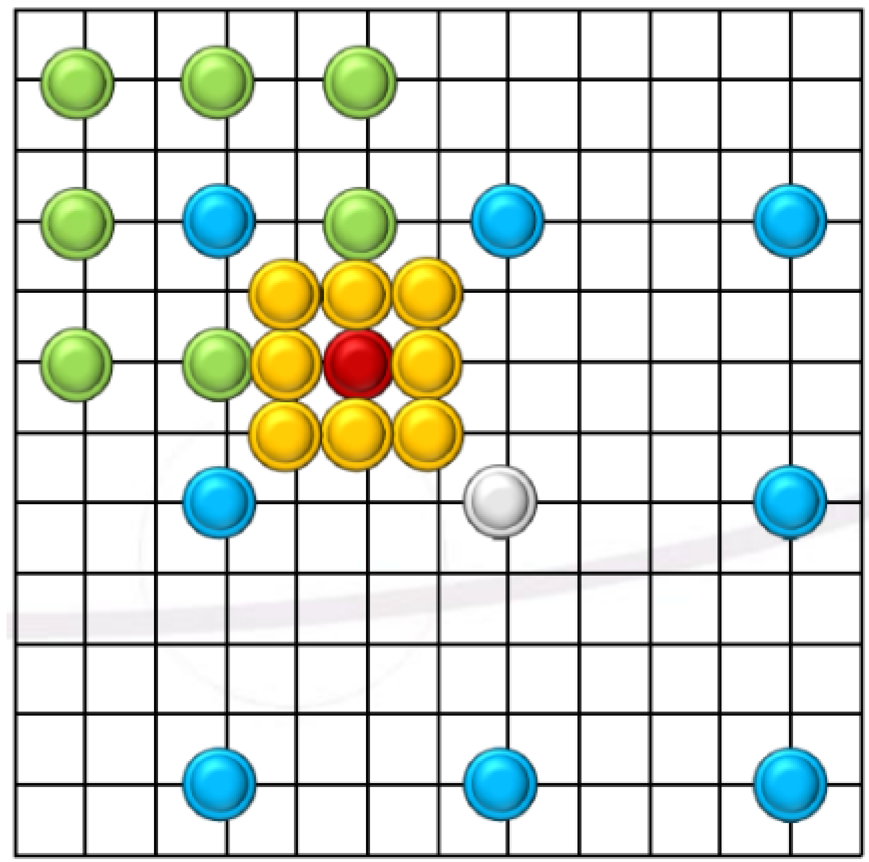
Block-based motion feature (2)

Searching criterion: For the reference image (block) X and the target image (block) Y, each has a size of $M \times N$

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(X(i,j) - Y(i,j))^2$$

Searching strategy

- Full (slow) search
- Sub-optimal (fast) search, e.g., three-step search shown in right figure

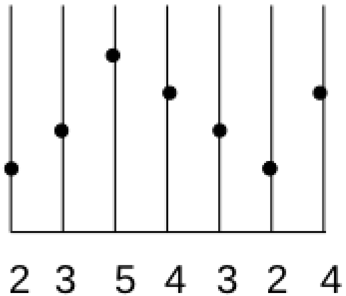


Block-based motion feature (3)

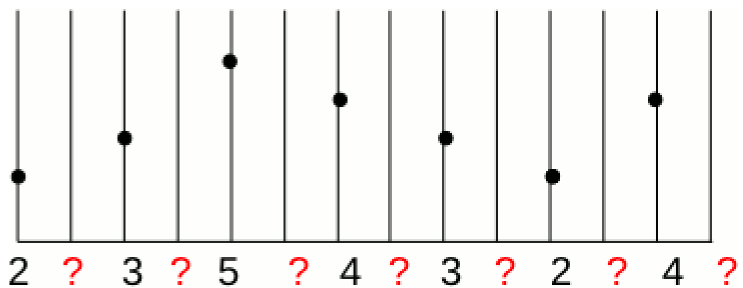
Searching precision
- Integer pixel precision
- Sub-pixel precision

- How do we compute the values of pixels at fractional positions?
- Image interpolation

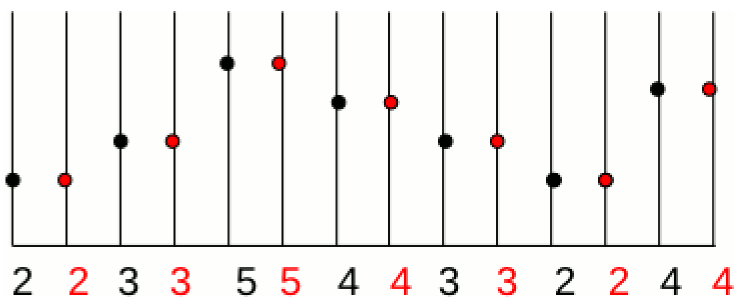
Toy example: input is 1D sequence



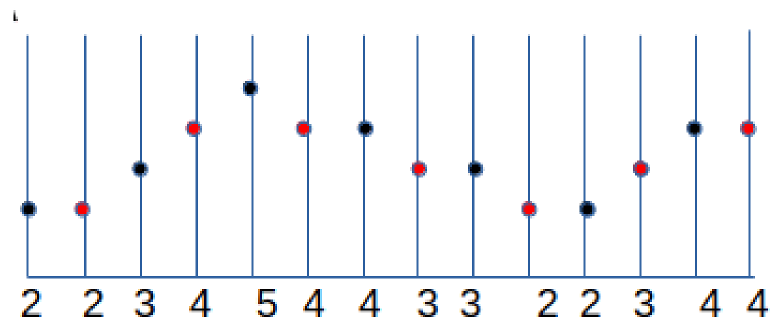
Output: interpolated 1D sequence



Output: nearest interpolation



Output: bilinear interpolation

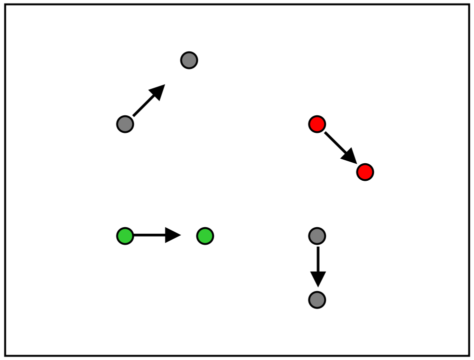


Reference: https://clouard.users.greyc.fr/Pantheon/experiments/rescaling/index-en.html

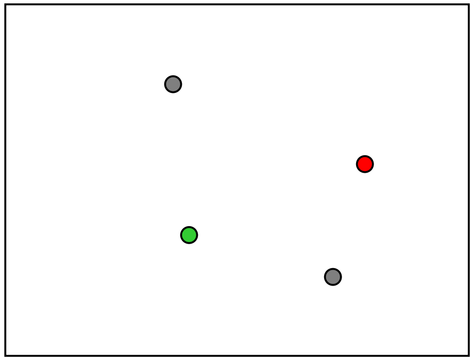
Optical flow (1)

- Definition: optical flow is the apparent motion of brightness patterns in the image
- Objective: Estimate image motion at each pixel from optical flow.

Example: Given two subsequent frames, estimate the motion vector $[u, v]$ at the pixel location (x, y).



$I(x, y, t)$



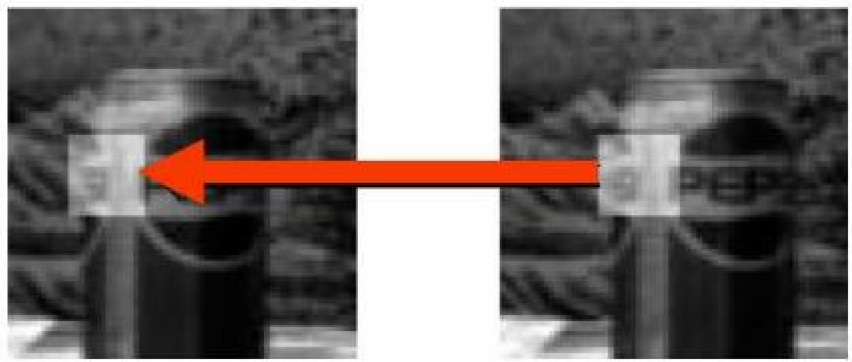
$I(x, y, t+1)$

Notation	
(x, y)	Pixel coordinate
I	Image
t	Frame index
$[u, v]$	Motion vector

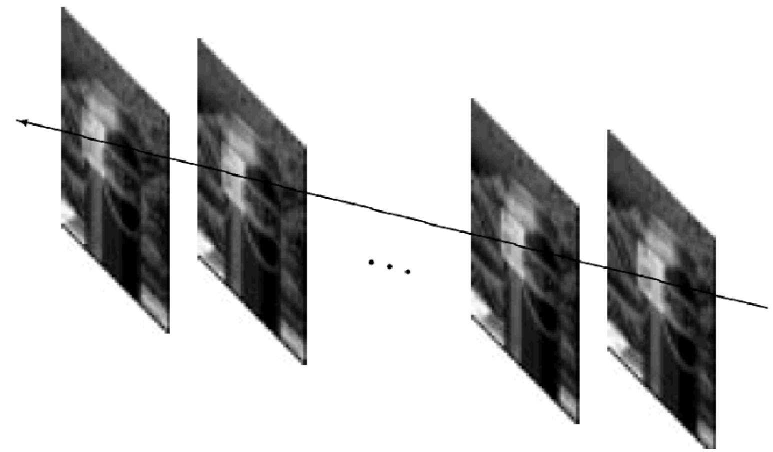
Optical flow (2)

Brightness consistency: Image brightness in a small region remain the same although their location may change. That is,

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



Small motion: The image motion of a patch changes gradually over time. (u, v are less than 1 pixel, and smoothly varying over the time)



Optical flow calculation (1)

- Brightness constancy: $I(x, y, t) = I(x + u, y + v, t + 1)$
- Small motion: We take Taylor series expansion of I

$$I(x+u, y+v, t+1) = I(x, y, t) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} + \cancel{\text{higher order terms}}$$

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

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

$$0 = I_x u + I_y v + I_t$$

Note: $I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}, I_t = \frac{\partial I}{\partial t}$

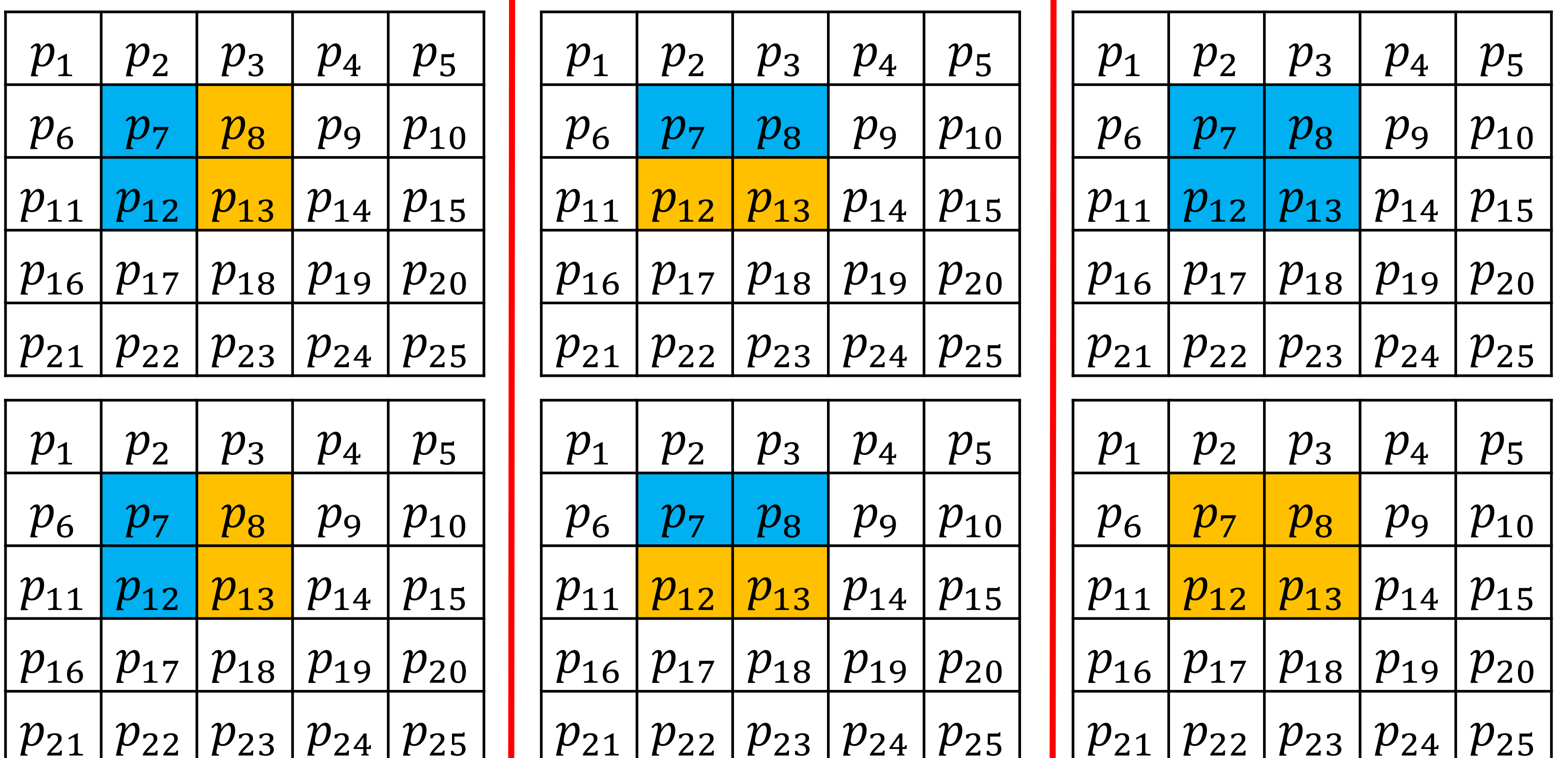
Spatial coherence constraint: Assume the pixel's neighbors have the same (u, v). If we use a 5×5 window centered at the pixel location (x, y), that gives us 25 equations for 25 pixels $p_1, p_2, \cdots, p_{25}$.

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix}$$

Optical flow calculation (2)

Spatial coherence constraint: Assume the pixel's neighbors have the same (u, v). If we use a 5×5 window centered at the pixel location (x, y), that gives us 25 equations.

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix}$$



Frame (t)

Frame $(t+1)$

$I_x(p_{13}) = \sum(\text{orange pixel} - \text{blue poixel})$

$I_y(p_{13}) = \sum(\text{orange pixel} - \text{blue poixel})$

$I_t(p_{13}) = \sum(\text{orange pixel} - \text{blue poixel})$

Other gradient calculation methods also can be used here.

Optical flow calculation (3)

Question: How can we use discontinuous frames (e.g., frame skipping)?

- Recall our small motion assumption

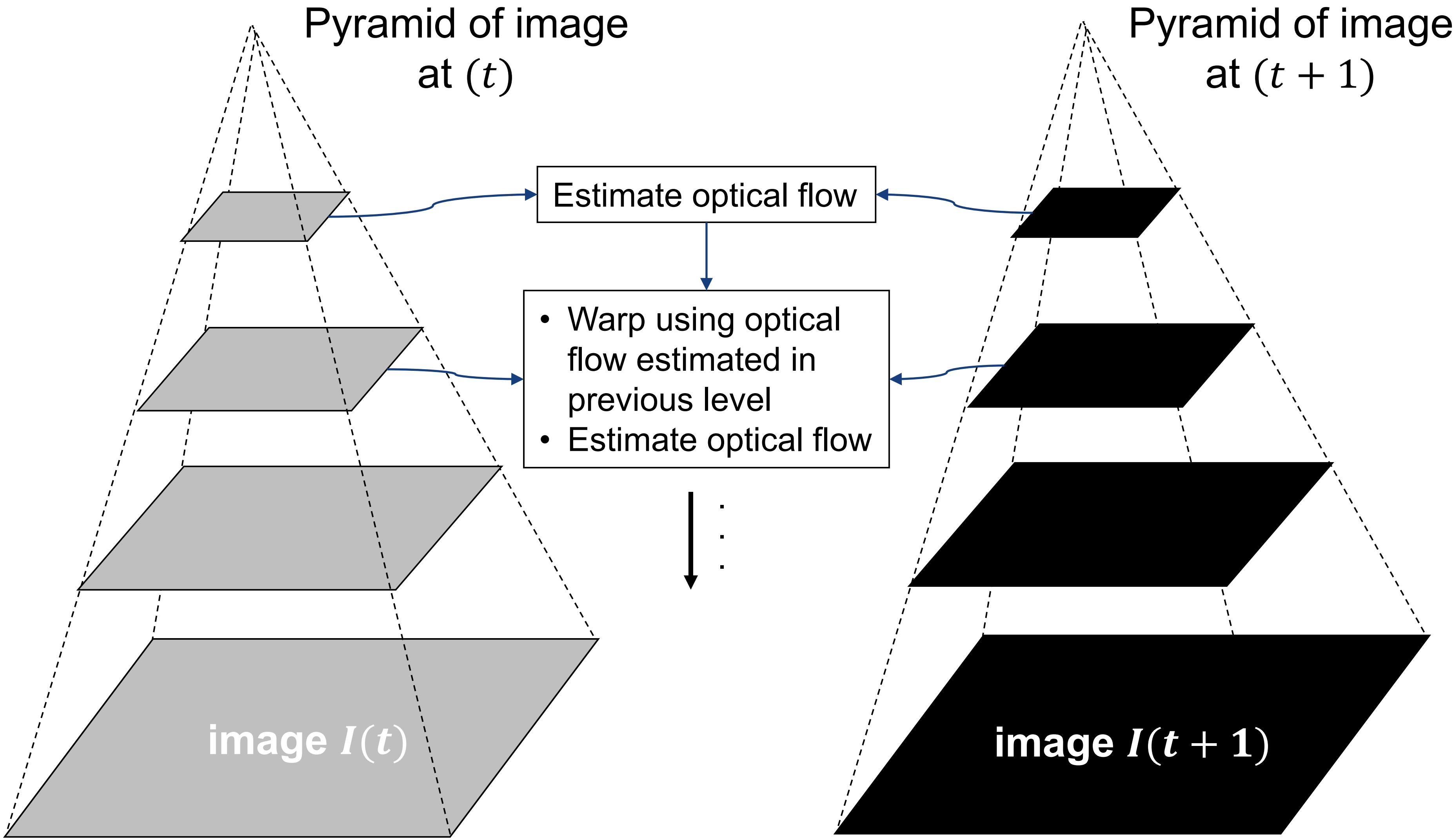
$$
\begin{aligned}
I(x+u, y+v) &= I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \cancel{\text{higher order terms}} \\
&\approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v
\end{aligned}
$$

- To do better, we need to add higher order terms back

Iterative Lucas-Kanade estimation

1. Estimate optical flow at each pixel by solving Lucas-Kanade equations
2. Warp $I(t)$ towards $I(t+1)$ using the estimated flow field
3. Repeat until convergence

Optical flow calculation (4)



Reference: B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision,". Int. Joint Conf. on Artificial Intelligence, pp. 674–679, 1981.

When/where optical flow fails?

- When: Can we differentiate the optical flow due to camera motion, lighting change, object motion? [No]
- Where: Which part of the flow vector cannot be determined from images? [flat regions of the image, or movement along the edge direction.]

Recall the optical flow solution for a 5×5 image patch

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix} \qquad \underset{25\times 2}{\mathbf{A}} \quad \underset{2\times 1}{\mathbf{d}} = \underset{25\times 1}{\mathbf{b}}$$

Least squares solution for $\mathbf{d}$ given by $(\mathbf{A}^T\mathbf{A})\mathbf{d} = \mathbf{A}^T\mathbf{b}$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

- Question: Where is $\mathbf{d}$ solvable?

$\mathbf{A}^T\mathbf{A}$ should be invertible, $\mathbf{A}^T\mathbf{A}$ should not be too small due to noise, eigenvalues λ_1 and λ_2 of $\mathbf{A}^T\mathbf{A}$ should not be too small.

- A texture image region: Yes
- A flat image region (e.g., plain wall): No

The summations in above equation are over all 25 pixels.

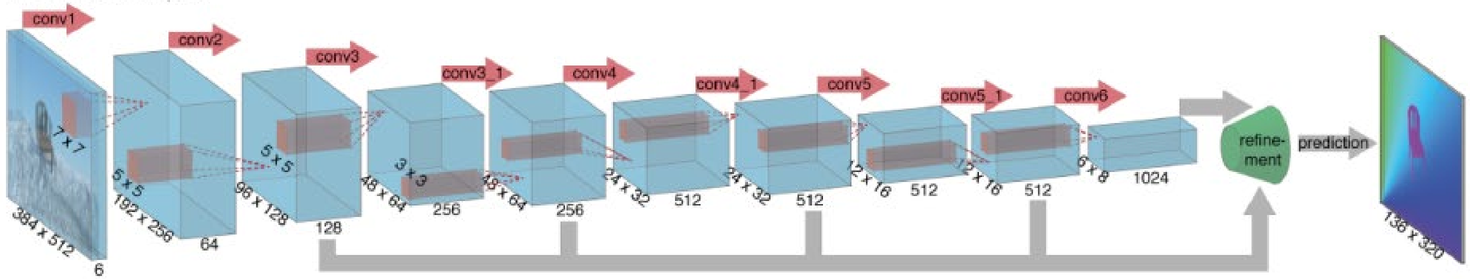
Optical flow using deep learning

- FlowNetSimple: Stack two sequentially adjacent input images together and feed them through the network.
- FlowNetCorr: First produce representations of the two images separately, and then combines them together in the 'correlation layer' (see next slide), and learn the higher representation together.

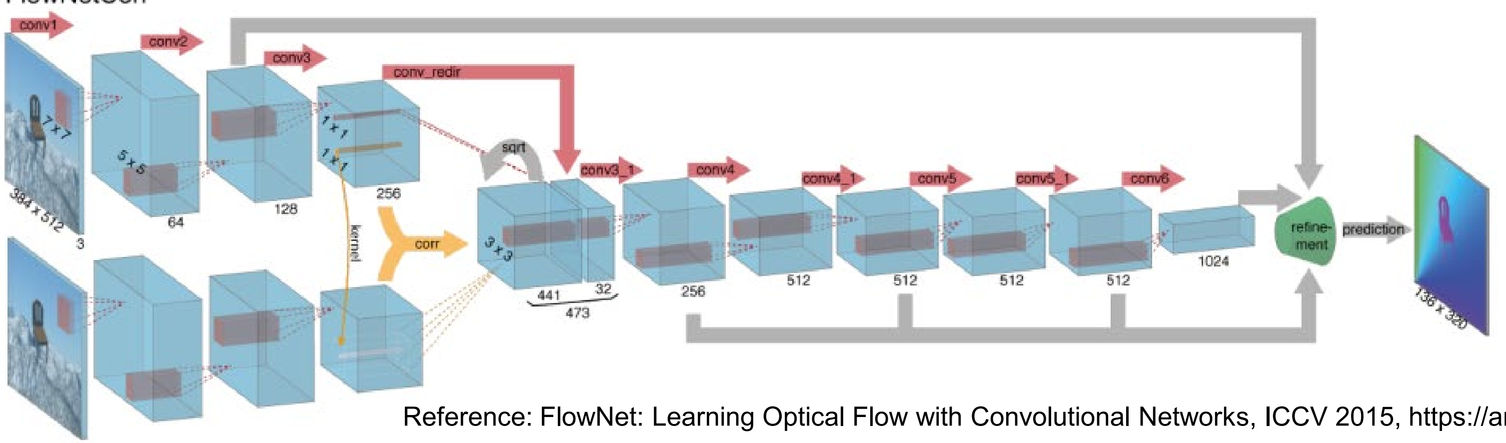
Training dataset

	Frame pairs	Frames with ground truth
Middlebury	72	8
KITTI	194	194
Sintel	1,041	1,041
Flying Chairs	22,872	22,872

FlowNetSimple



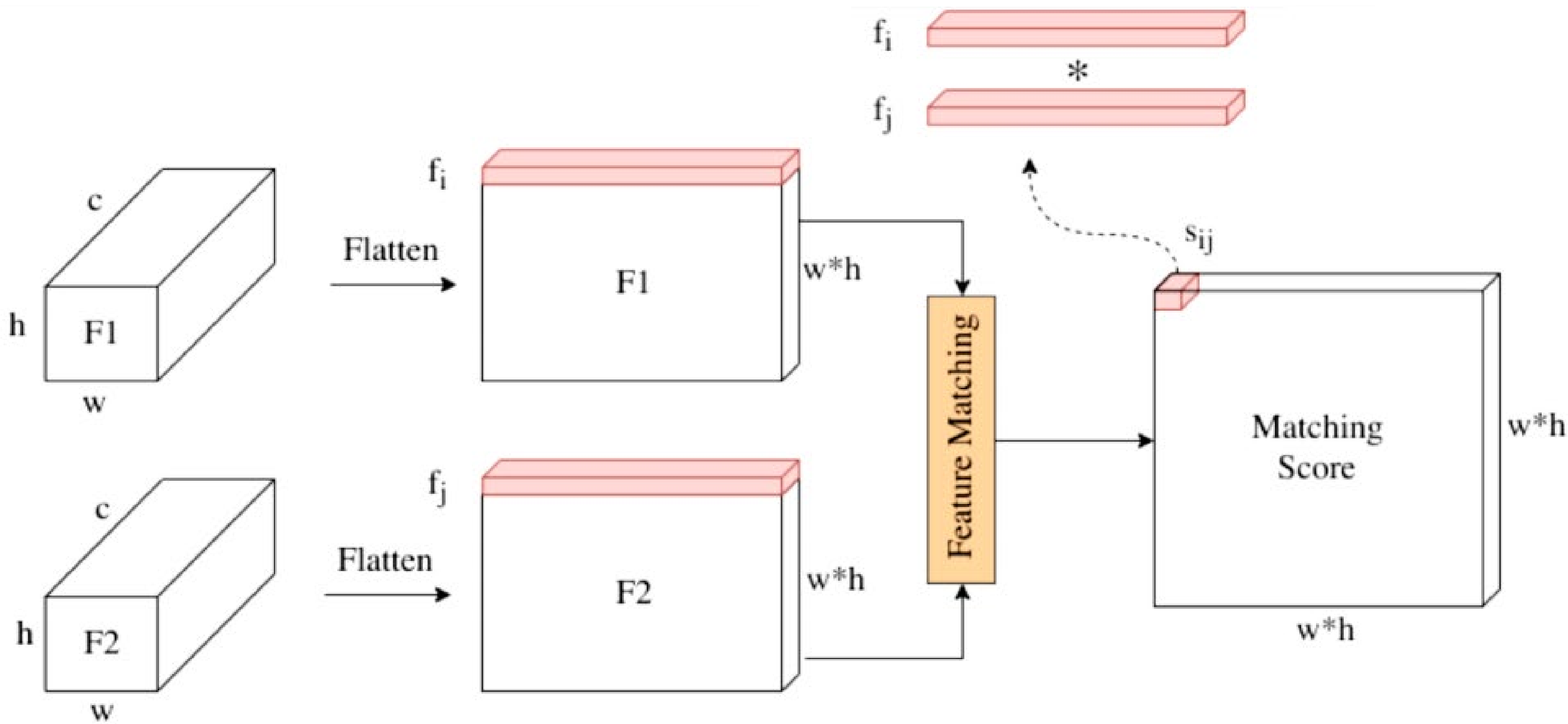
FlowNetCorr



Reference: FlowNet: Learning Optical Flow with Convolutional Networks, ICCV 2015, https://arxiv.org/abs/1504.06852

Optical flow using deep learning

- Correlation layer: Dot product between pair-wise feature vectors.



Reference: FlowNet: Learning Optical Flow with Convolutional Networks, ICCV 2015, https://arxiv.org/abs/1504.06852

Usages of motion features

- Tracking objects
- Segmenting objects based on motion cues
- Recognizing events and activities
- Motion-based video generation

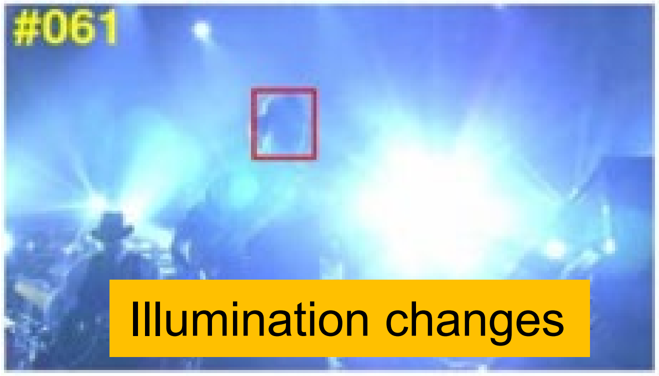
Demo website: https://monalisaeffect.com/

- Photo: https://www.softwebsolutions.com/resources/video-analytics-use-cases.html
- Reference: Youtube large scale video understanding, https://research.google.com/youtube8m/workshop2017/index.html;

Topics

- Fundamentals of video data modelling, and motion feature representation methods
- Object tracking in the video
- Workshop: Build motion feature extraction and object tracking in the video

Challenges in tracking



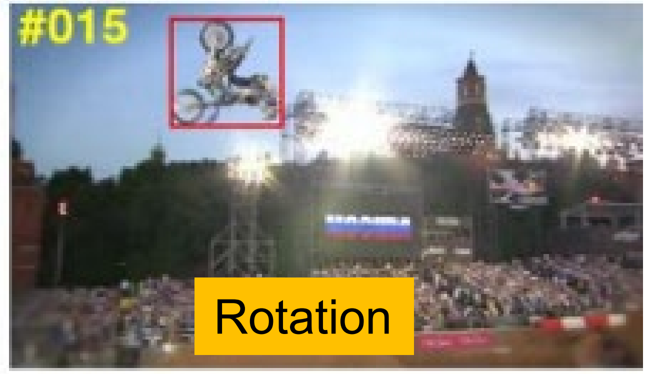
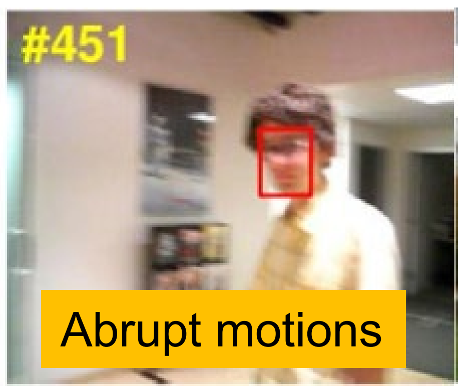
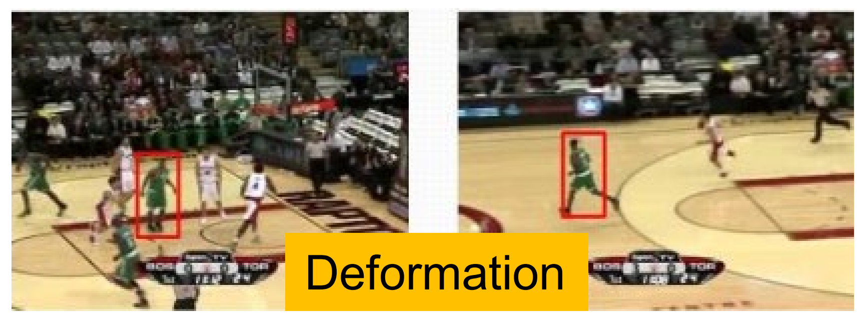


Photo:
https://pdfs.semanticscholar.org/9eda/66f8551e20fca3b1f9275781e9555ea5023d.pdf





Detection vs tracking

Detection	Tracking
Applied on static image, or independently on each frame of the sequence	Applied on a sequence of images
Relies on detector trained from (known) object data off line.	Can handle (unknown) object on line.
Can get the position of objects	Preserve identity, can get the trajectory of the objects
Can not handle occluded objects	Can handle occluded objects (based on tracking assumption)

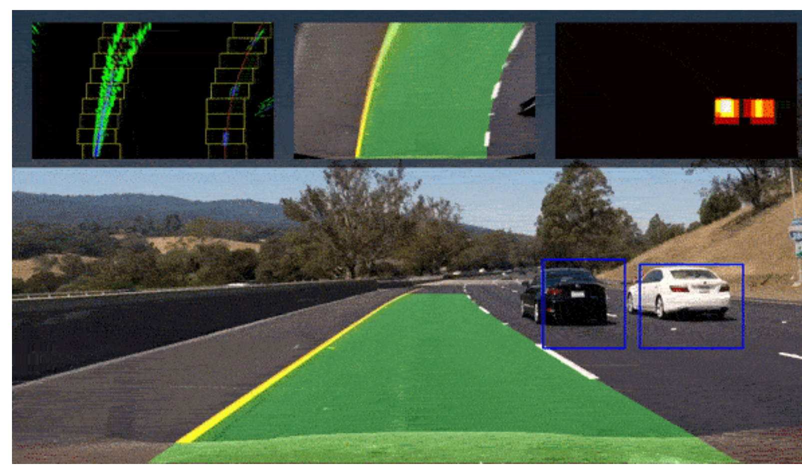


Photo: http://vision.stanford.edu/teaching/cs131_fall1819/index.html

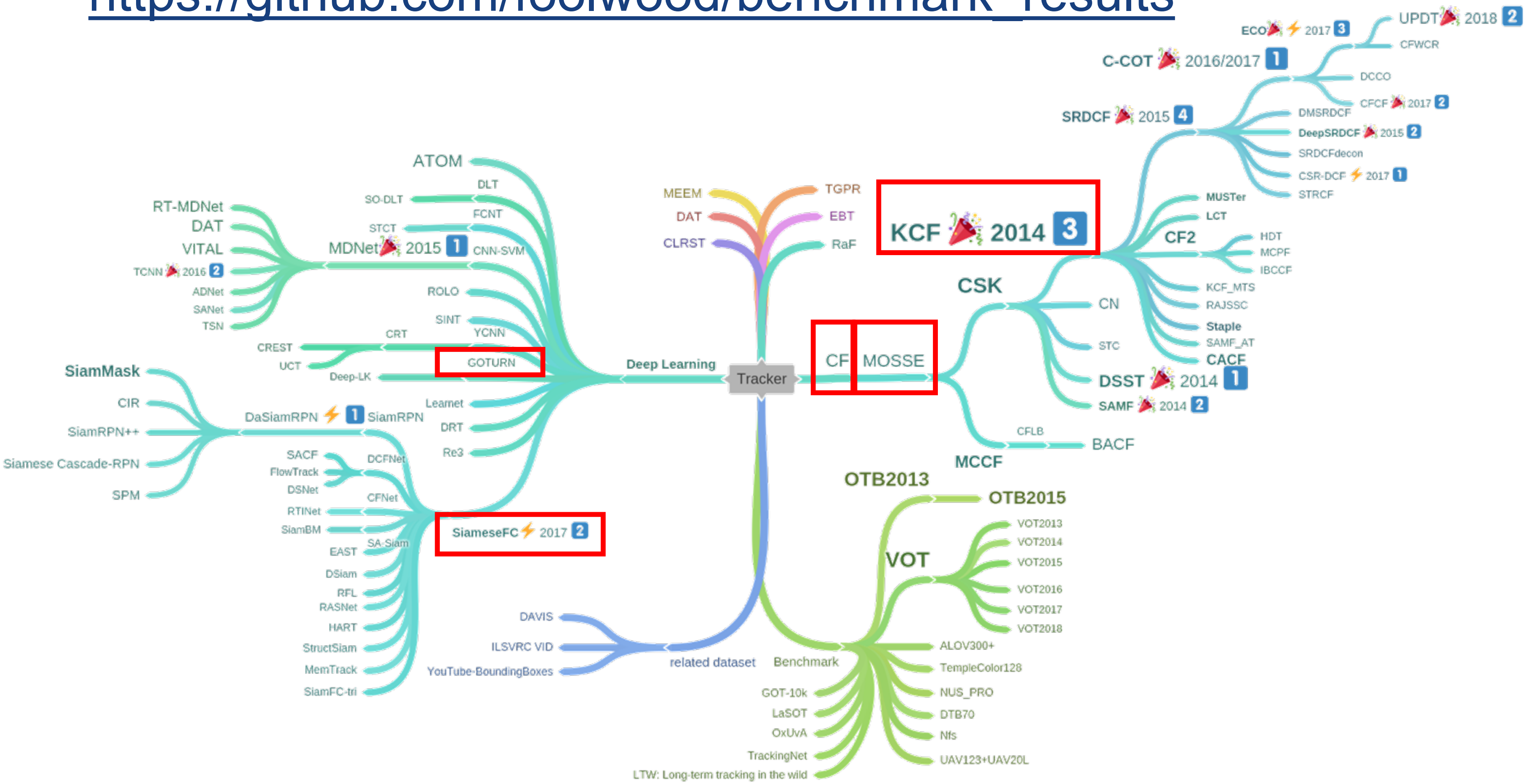
Key components of object tracking

- Tracker initiation: We need to define the initial state of the target by either (manually) drawing a bounding box or (automatically) detecting an object of interest (say, human, car, etc).
- A module holding the internal representation of the object: We need to learn the visual appearance of the object.
- A module observing the features of the search image (candidate regions).
- A module evaluating the object and the search image for estimating the motion: Evaluate the candidates in a searching region to identify the exact location of the target.
- A module updating the representation of the object: The object might change its appearance during tracking.

- Generative methods: Represent objects with appearance models, and track targets by searching for the image region most similar to the models.
- Discriminative methods: Treat objects tracking as a binary classification problem to distinguish the target from non-target background.

Tracking benchmark

- Object tracking benchmark, https://github.com/foolwood/benchmark_results



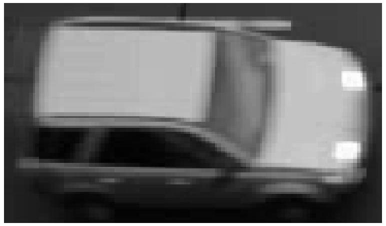
Generative: Template matching

- Simple color similarity based template matching.
- The function slides through the image I, compares the overlapped patches of size $w \times h$, against the object template T, using the specified method and stores the comparison result.

At the position (x, y), the score $R(x, y) = \sum_{m,n}(T(m, n) - I(x + m, y + n))^2$

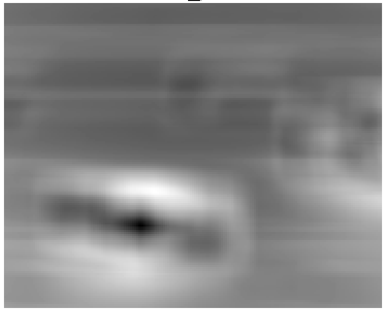
cv2.TM_SQDIFF: Absolute difference

Target object template



Target object in frame 1

Matching score map (cv2.TM_SQDIFF)



Detected object position in frame 2

- Good for matching the same type of objects with similar appearance
- Not work well for the variations in pose, lighting, natural variations or non-rigid transformations.

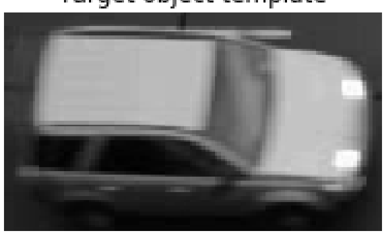
Generative: Template matching

- Simple cross correlation based template matching.
- The function slides through the image I, compares the overlapped patches of size $w \times h$, against the object template T, using the specified method and stores the comparison result.

At the position (x, y), the score $R(x, y) = \sum_{m,n}(T(m, n) \times I(x + m, y + n))$

cv2.TM_CCORR: Cross correlation

Target object template



Target object in frame 1

Matching score map (cv2.TM_CCORR)

Detected object position in frame 2

Generative: Template matching

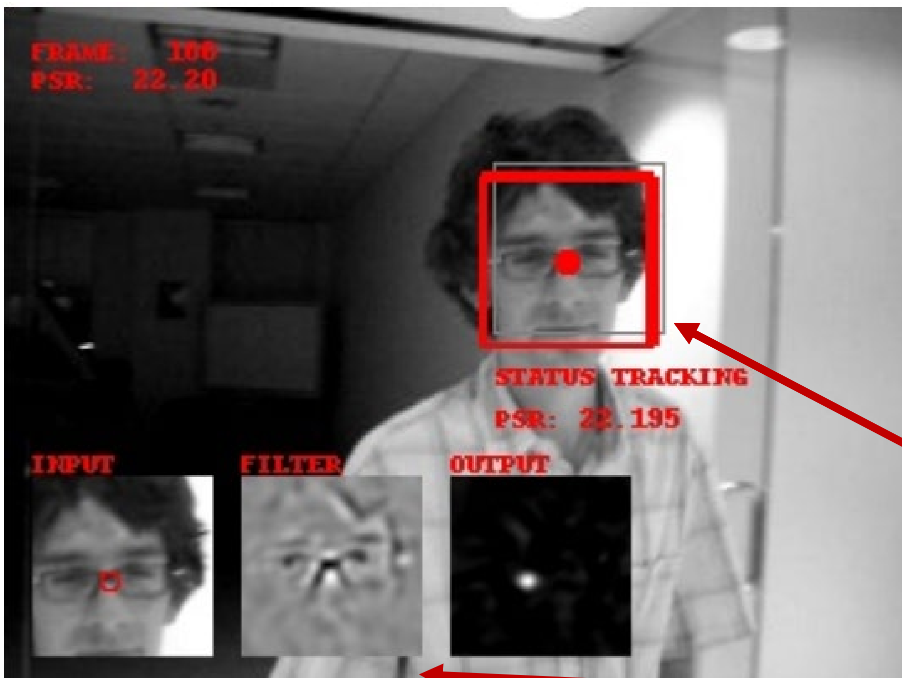
Challenge in template matching

- Time consuming: Repeat this correlation calculation on the whole image.

→ Perform checking in the search region only (not the whole image).

- Not robust to illumination change: Correlation function relies on intensity.

→ Learn an optimized filter (see next slide).



Note:

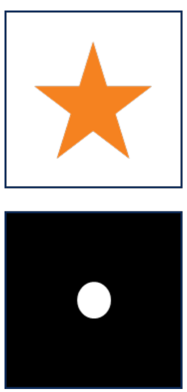
- Search region (in the current frame) is usually slightly larger than the target region in the previous frame.
- Filter size is as same as search region size.

Reference: Bolme, et. al., Visual object tracking using adaptive correlation filters, http://www.cs.colostate.edu/~vision/publications/bolme_cvpr10.pdf

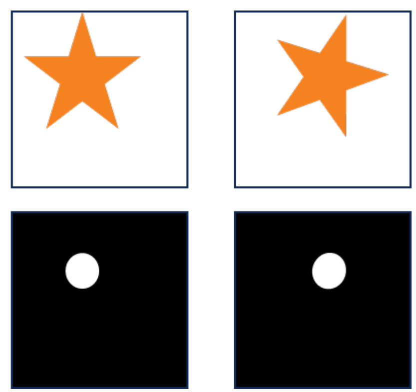
Correlation filter: MOSSE

- Idea: Learn an "optimized correlation filter", when correlated with a target template, produces a strong peak at the location of the object and low (or even zero) values at all the other locations. (More robust than a single template-based tracking.)
- Minimum output sum of squared error filter (MOSSE) filter minimizes the sum of squared error between the actual output of the correlation and the desired output of the correlation. $w^* = \underset{w}{\operatorname{argmin}}(\sum_i \|x_i \circledast w - y_i\|^2)$. Given a target template and a synthetic response image contains a Gaussian peak centred on the object. Apply affine transformation (e.g., transition, rotation) to generate several pairs of x_i (synthetic input image) and y_i (synthetic response image) to calculate w (unkown optimized filter).
- Kernelized Correlation Filters (KCF): Further introduce regularization, kernel matrix, into the cost function, such as $w^* = \underset{w}{\operatorname{argmin}}(\sum_i \|x_i \circledast w - y_i\|^2 + \text{Regularization } (w))$

1. Given the target and desired response



2. Generate more pairs of synthetic images and responses



3. Calculate the optimized filter
4. Apply the filter on the test region to obtain the response map

Reference:

- Bolme, *et. al.*, Visual object tracking using adaptive correlation filters, CVPR, 2010, http://www.cs.colostate.edu/~vision/publications/bolme_cvpr10.pdf
- MOSSE demo code: https://github.com/TianhongDai/mosse-object-tracking
- J. F. Henriques, *et al.*, High-Speed Tracking with Kernelized Correlation Filters, IEEE Trans on PAMI, Vol. 37, No. 3, 2015, pp. 583-596.

Discriminative tracking

- Train an online discriminative classifier based on positive and negative samples.

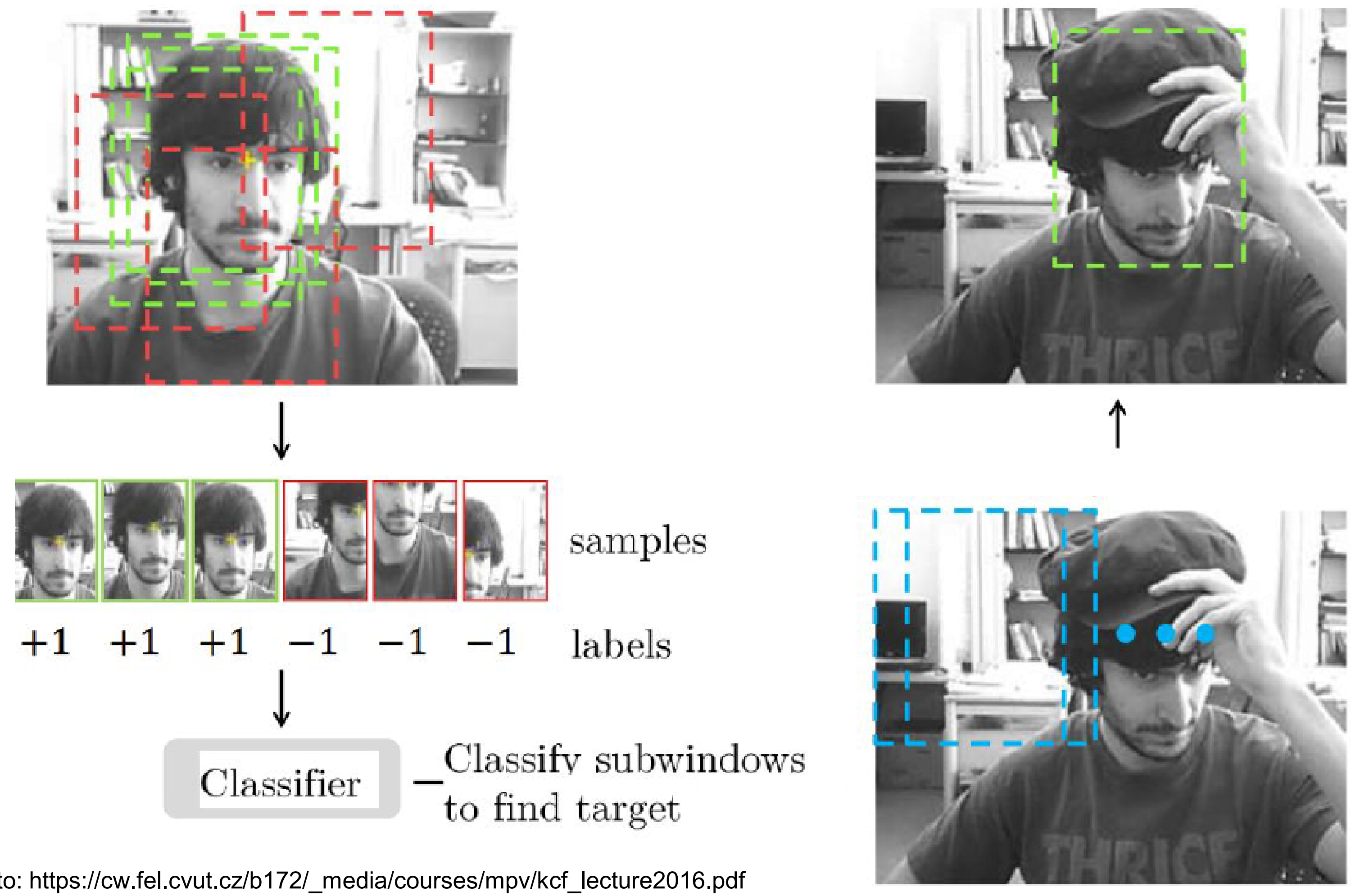


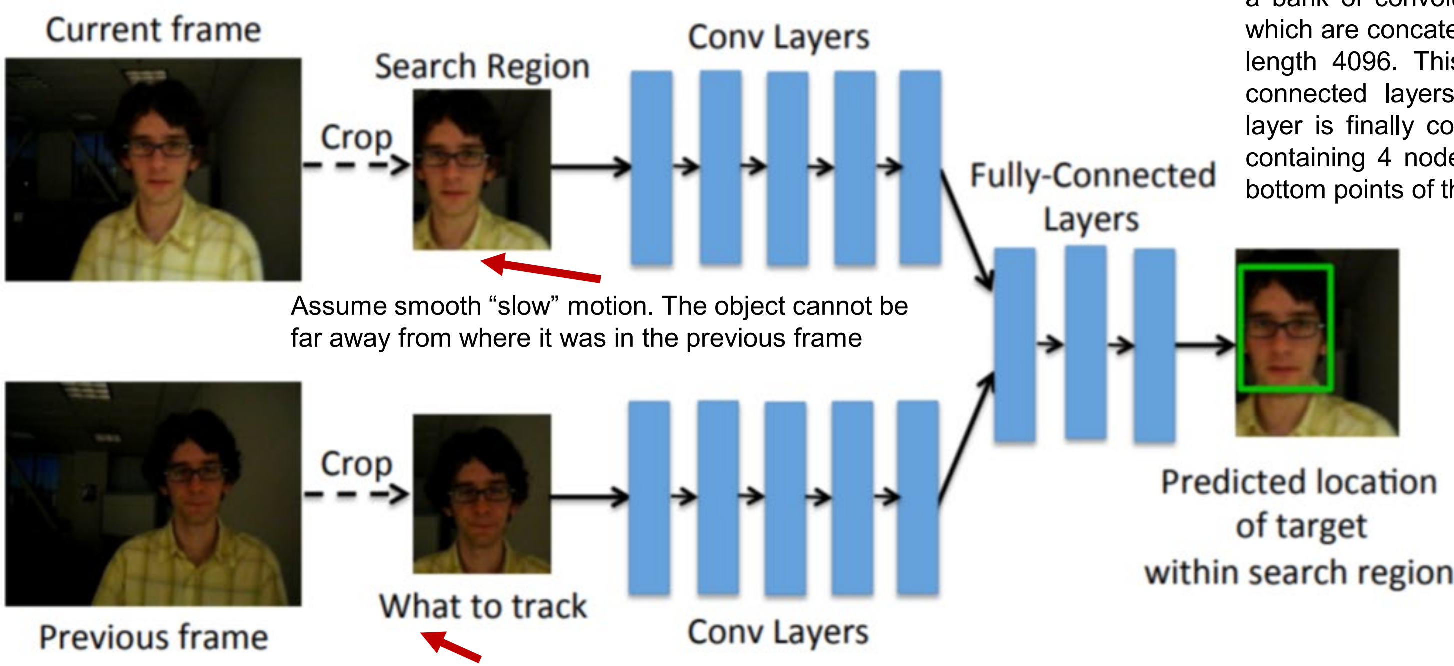
Photo: https://cw.fel.cvut.cz/b172/_media/courses/mpv/kcf_lecture2016.pdf

Tracking using deep learning: GOTURN

GOTURN: Generic Object Tracking Using Regression Networks

- Input: A search region (candidates) from the current frame and a target from the previous frame
- Output: The coordinates (regression) of the object in the current frame, relative to the search region. The network's output consists of the coordinates of the top left and bottom right corners of the bounding box.

Model details: Both input frames pass through a bank of convolutional layers, the output of which are concatenated into a single vector of length 4096. This vector is input to 3 fully connected layers. The last fully connected layer is finally connected to the output layer containing 4 nodes representing the top and bottom points of the bounding box.



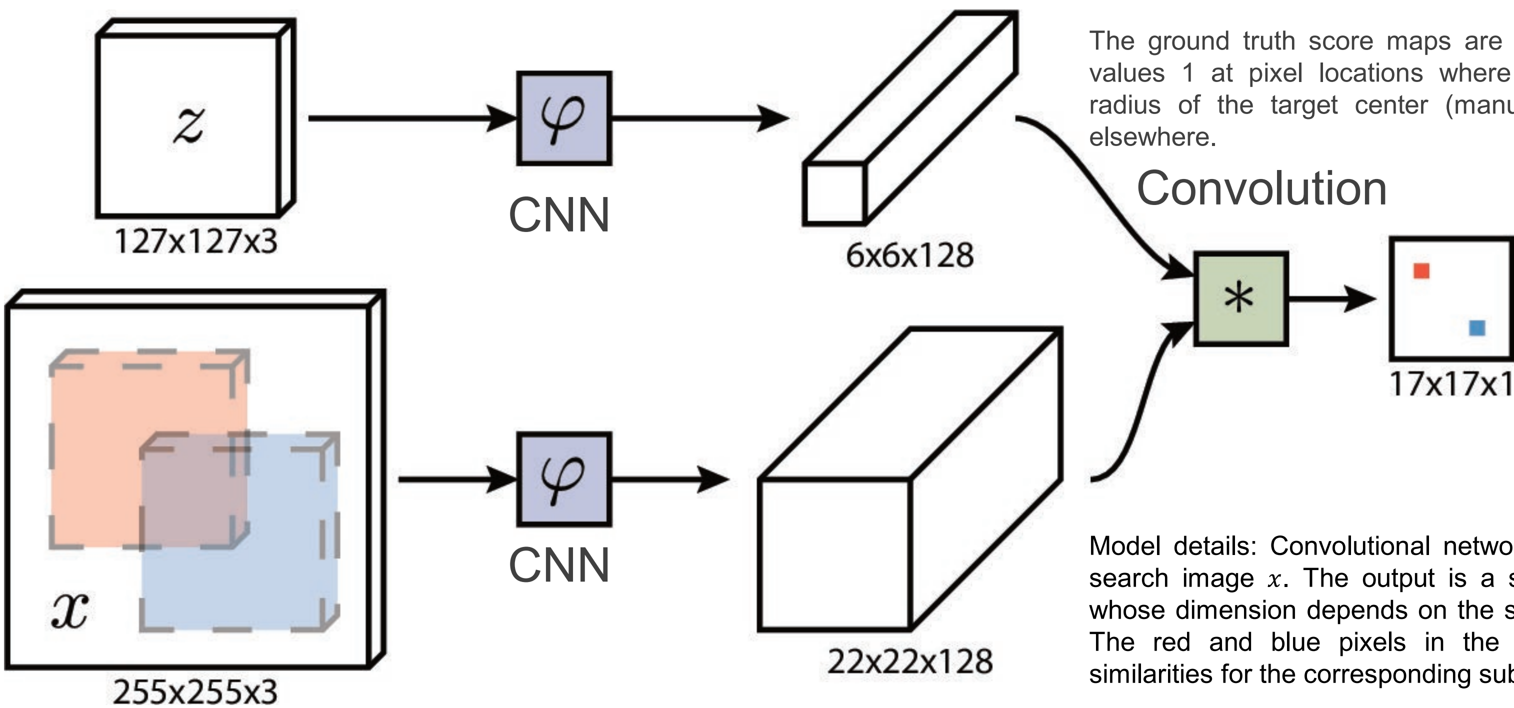
Assume smooth "slow" motion. The object cannot be far away from where it was in the previous frame

Crop the object to be tracked: Initialization of our tracker

Reference: D. Held, S. Thrun, S. Savarese, Learning to Track at 100 FPS with Deep Regression Networks, ECCV 2016, http://davheld.github.io/GOTURN/GOTURN.html

Tracking using deep learning: SiamFC

- Input: A target (in the previous frame, z) and a search region (candidates, x) centered at the previous position of the target. The dimensions (e.g., $127 \times 127 \times 3$) used in this example is just for illustration.
- Output: A score map. The position of the maximum score relative to the center of the score map, multiplied by the stride of the network, gives the displacement of the target from frame to the frame.



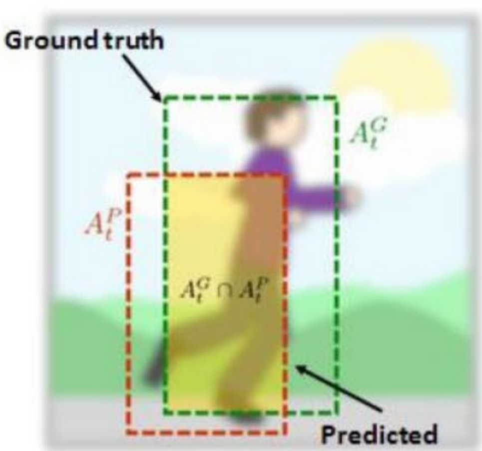
The ground truth score maps are automatically generated: values 1 at pixel locations where they are within certain radius of the target center (manually labelled); values 0 elsewhere.

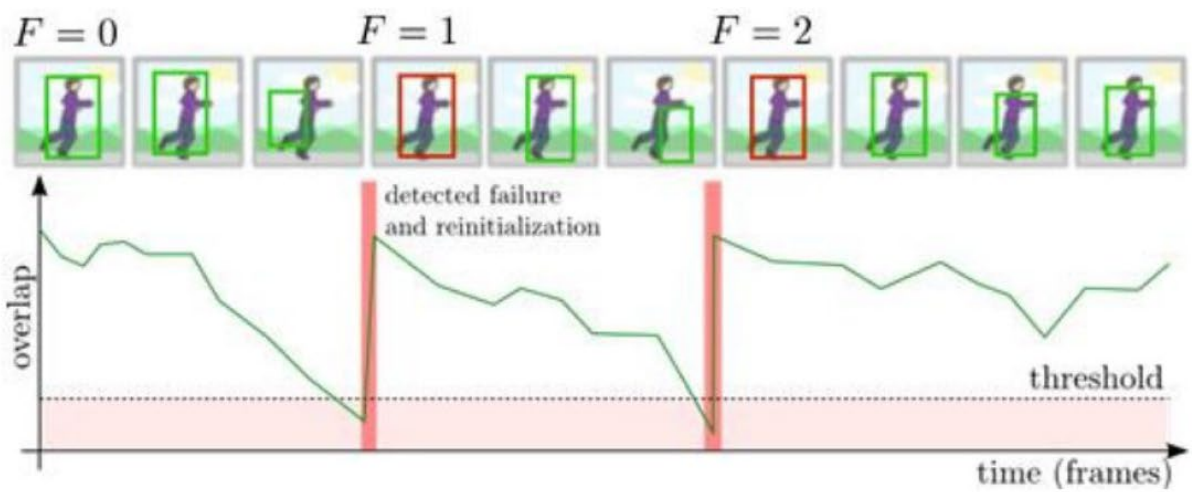
Model details: Convolutional network φ with respect to the search image x. The output is a scalar-valued score map whose dimension depends on the size of the search image. The red and blue pixels in the score map contain the similarities for the corresponding sub-windows.

Reference: Fully-convolutional Siamese networks for object tracking, https://www.robots.ox.ac.uk/~luca/siamese-fc.html

Single object tracking evaluation metric

- Accuracy: Average overlap during successful tracking
- Robustness: Number of times a tracker drifts off the target
- One Pass Evaluation: Run tracker throughout a test sequence initialized by ground truth bounding box in the first frame and return the average precision.
- Spatial Robustness Evaluation: Run tracker throughout a test sequence with initialization from a few different bounding boxes by shifting or scaling ground truth in the first frame and return the average precision.





Reference: Cehovin, et al., "Visual object tracking performance measures revisited," IEEE Trans. on Image Processing, Vol. 25, No. 3, 2016, pp. 1261-1274, https://arxiv.org/pdf/1502.05803.pdf

Multiple object tracking

- For each frame, first localize all objects using an object detector
- Associate detected objects between frames
- Make multiple object tracking to be a association problem more than a tracking problem.

We have N objects in previous frame and M objects in current frame. We can build a table of match scores $m(i,j)$ for $i=1,2,\cdots,N$ and $j=1,2,\cdots,M$.

The pairwise match score between objects in two consecutive frames

	A	B	C	D	E
A	0.95	0.76	0.62	0.41	0.06
B	0.23	0.46	0.79	0.94	0.35
C	0.61	0.02	0.92	0.92	0.81
D	0.49	0.82	0.74	0.41	0.01
E	0.89	0.44	0.18	0.89	0.14

Q1: How to obtain match score?

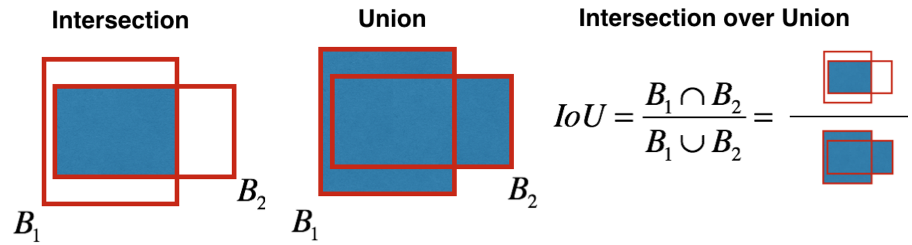
- Association based on location, motion, appearance and so on. (see following slides)

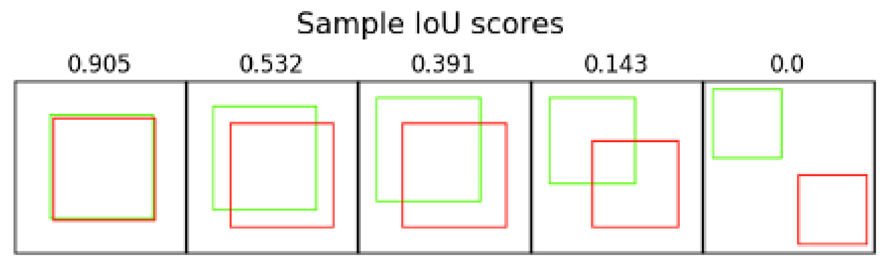
Q2: How to choose 1-1 correspondence that maximizes sum of match scores?

- We have many methods to address this issue, such as Hungarian algorithm, see more methods in the "Vision-based tracking" course notes http://www.cse.psu.edu/~rtc12/CSE598C/datassocPart1.pdf

Location model

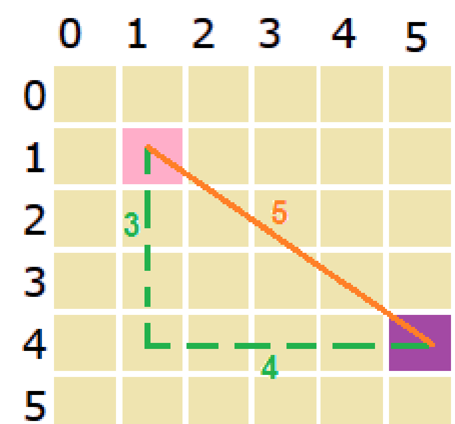
- The overlapping between the detected in current frame with that of the previous frame
- If there is no overlapping, we can calculate various distances based on center of object position.
- The object positions are indicated by their bounding box.





Euclidean distance: $\sqrt{(5-1)^2 + (4-1)^2} = 5$

Manhattan distance: $|5-1| + |4-1| = 7$



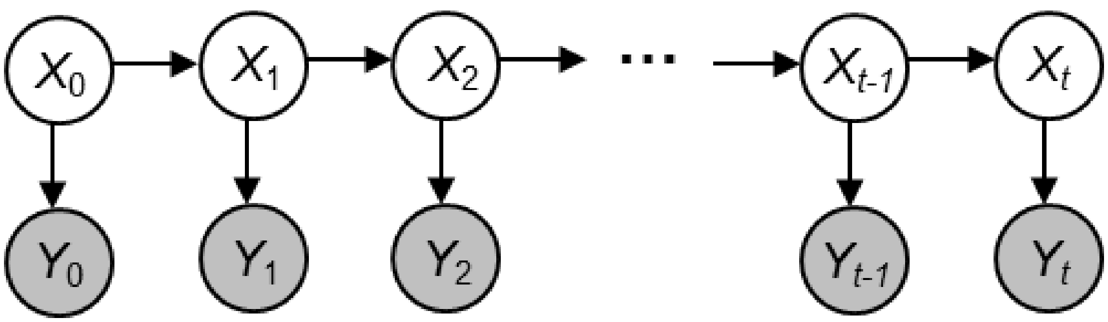
Reference: IOU, http://ronny.rest/tutorials/module/localization_001/iou/

Motion model

Model the movement of objects.

- State X: The actual state of the moving object that we want to estimate but cannot observe
  - E.g. center position of the object, aspect ratio a, width w, height h and their respective $velocities$ of the bounding box.
- Observations Y: Our actual measurement or observation of state X, which can be very noisy
- At each time t, the state changes to X_t to have a new observation Y_t
- Our goal is to recover the most likely state X_t given:
  - All observations so far, i.e. $Y_1, Y_2, \cdots, Y_t$
  - Knowledge about dynamics of state transitions

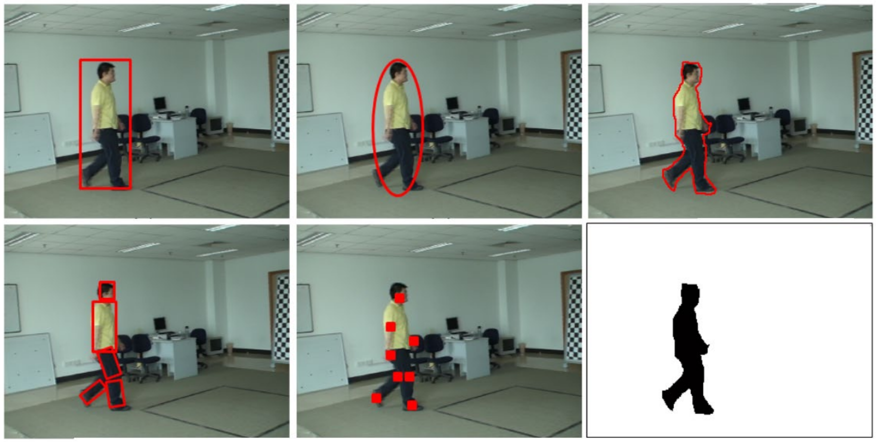
We can use Kalman filter or particle filter. (out of the scope of this course)



Appearance model

Model the appearance of the object to evaluate the similarities.

- What to track?
  - Bounding box, ellipse, contour, interest point, and silhouette
- What visual representations are appropriate and robust for visual object tracking?
  - Hand-crafted features like histograms and color, CNN features
- Which similarity evaluation schemes are suitable for visual object tracking?
  - Techniques in single object tracking like cross correlation and SiameseFC can be used here



Reference: A Survey of Appearance Models in Visual Object Tracking, https://arxiv.org/abs/1303.4803

Multiple object tracking: Integrated

Example: DEEP SORT

- Track initialization (e.g., using a deep learning-based detector)
- Matching detections at two consecutive time stamps
- Repeat for every pair of frames

Reference:

- Simple Online and Realtime Tracking with a Deep Association Metric, https://arxiv.org/abs/1703.07402
- https://medium.com/@riteshkanjee/deepsort-deep-learning-applied-to-object-tracking-924f59f99104

Multiple object tracking benchmark

MOT, https://motchallenge.net

- Pedestrian tracking, 7 training videos and 7 test videos

KITTI, http://www.cvlibs.net/datasets/kitti/eval_tracking.php

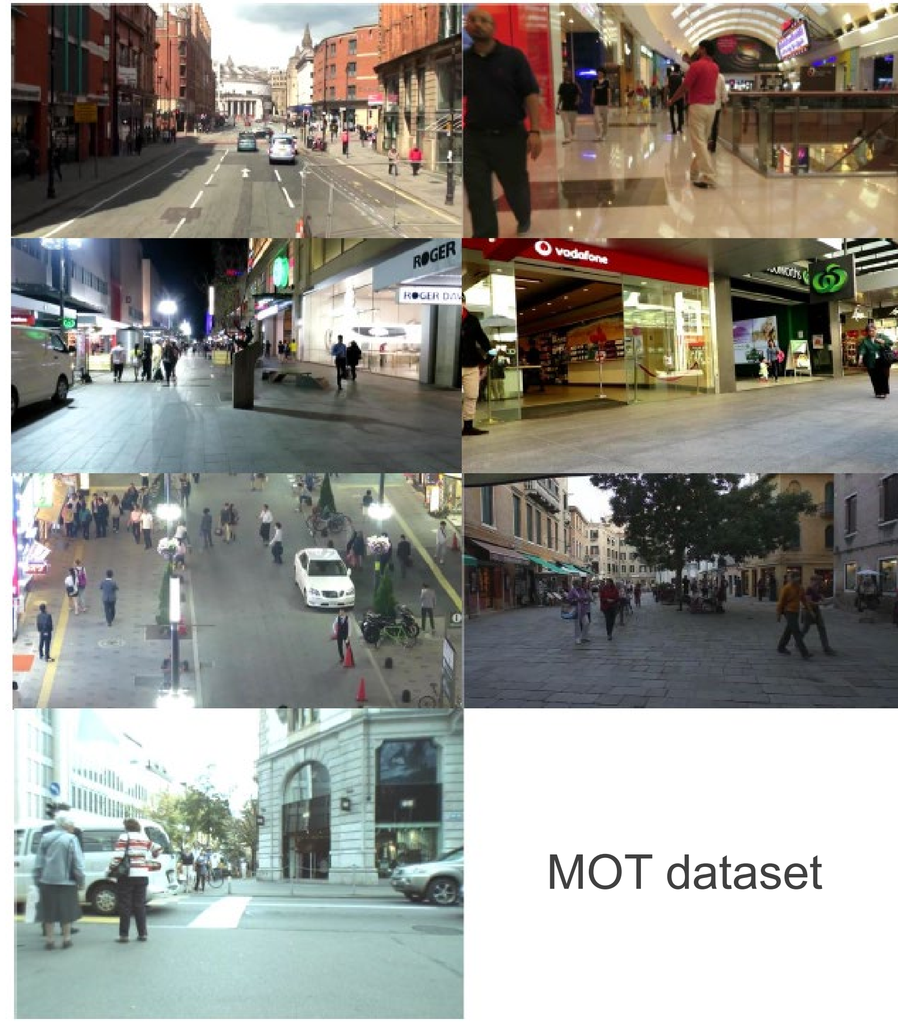
- Car and pedestrian tracking

ImageNet VID, http://image-net.org/challenges/LSVRC/2017/

- 30 classes

Evaluation metric: MOTA, it combines three error sources for every frame t, false negative fn_t, false positives fp_t, and identity switches id_sw_t.

$$MOTA = 1 - \frac{\sum_t (fn_t + fp_t + id\_sw_t)}{\sum_t g_t}$$



MOT dataset

Reference: MOT16: A benchmark for multi-object tracking, https://arxiv.org/abs/1603.00831

Practical issue of object tracking

- How to initialize the object in the first frame to activate the tracker?
  - Manual initialization
  - Object detector
- Where to find labelled data for training a object tracker?
  - Unfortunately, there is no ImageNet dataset in object tracking research domain.
  - Fortunately, we don't need to re-train the tracker, as the tracker should be general to handle various types of objects.
- Speed is extremely important requirement for real-time tracker.

Workshop

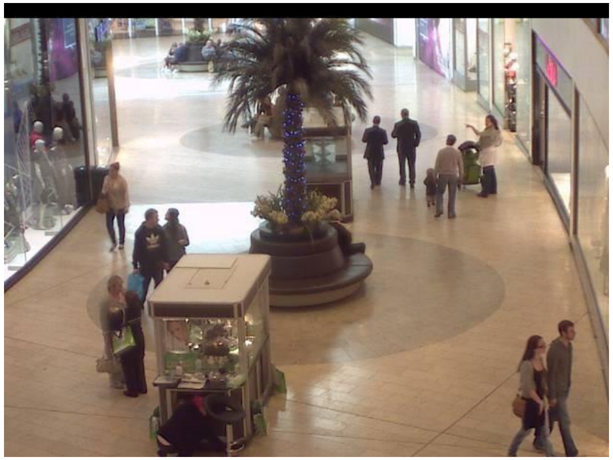
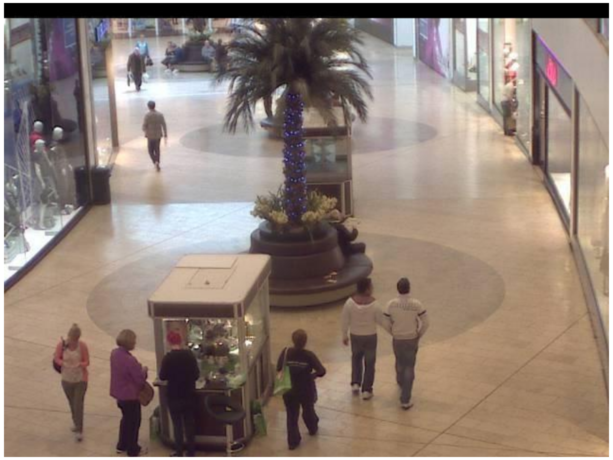
- Exercise 1: Motion analysis for indoor retail surveillance video
- Exercise 2: Object tracking for outdoor traffic monitoring surveillance video

Exercise 1: Indoor retail surveillance

"Complementing the transaction data from in-store smart terminals are the video analytics generated by CCTV cameras outside the stores, which capture footfall and demographic data throughout the mall and into each store." Quoted from Funan Mall.

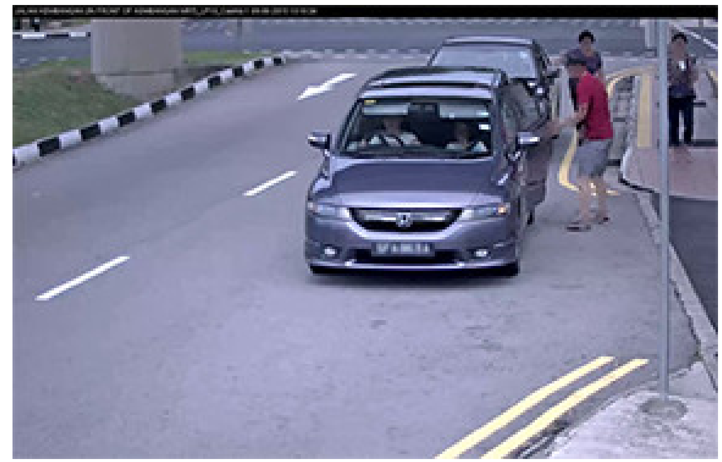
- Maximize store layout and navigation
- Optimize promotions and product displays
- Manage store traffic
- Streamline checkout
- Analyze consumer demographics

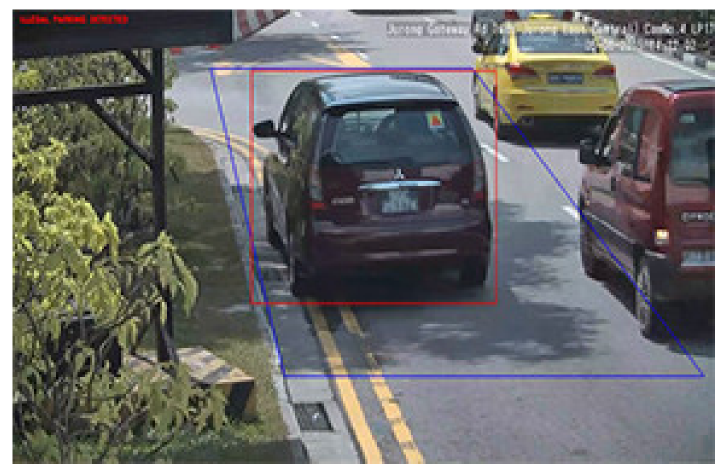
Reference:
- https://www.retailcustomerexperience.com/blogs/transforming-video-content-analytics-into-retail-business-intelligence/
- https://www.capitaland.com/international/en/about-capitaland/newsroom/news-releases/international/2018/may/nr-20180525-capitaland-shapes-funan-into-singapore-first-online-and-offline-shopping-mall.html

Exercise 2: Traffic monitoring

- Traffic management
- Illegal parking
- Privacy masking





Thank you!

Dr TIAN Jing
Email: tianjing@nus.edu.sg

