Tsinghua University

Advanced Computer Vision

THU×SENSETIME – 80231202

sensetime

## Chapter1 - Section 2

# Image and Video Processing

Dr. Dai Jifeng

Friday, February 26, 2021

# Outline

商汤 sensetime

清華大學 Tsinghua University

# Highlights

Image & video representation in computer

Basic applications of image processing

Traditional video processing and feature extraction methods

Common algorithms for image and video compression

History of digital image processing

# Outline

| | |
|---|---|
| Part 1 | Image and video representation |
| Part 2 | Image processing |
| Part 3 | Video processing |

# Image & video Representation

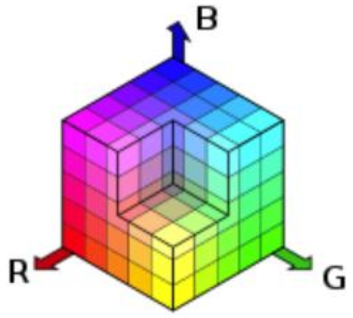
- Image -- A 2D discrete signal

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 111 | 115 | 113 | 111 | 112 | 111 | 112 | 111 |
| 135 | 138 | 137 | 139 | 145 | 146 | 149 | 147 |
| 163 | 168 | 188 | 196 | 206 | 202 | 206 | 207 |
| 180 | 184 | 206 | 219 | 202 | 200 | 195 | 193 |
| 189 | 193 | 214 | 216 | 104 | 79 | 83 | 77 |
| 191 | 201 | 217 | 220 | 103 | 59 | 60 | 68 |
| 195 | 205 | 216 | 222 | 113 | 68 | 69 | 83 |
| 199 | 203 | 223 | 228 | 108 | 68 | 71 | 77 |

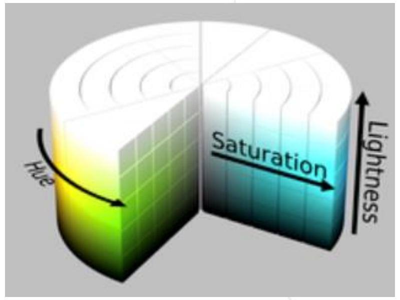
- Video -- Sequences of images

time

- **Color Model**
  - RGB

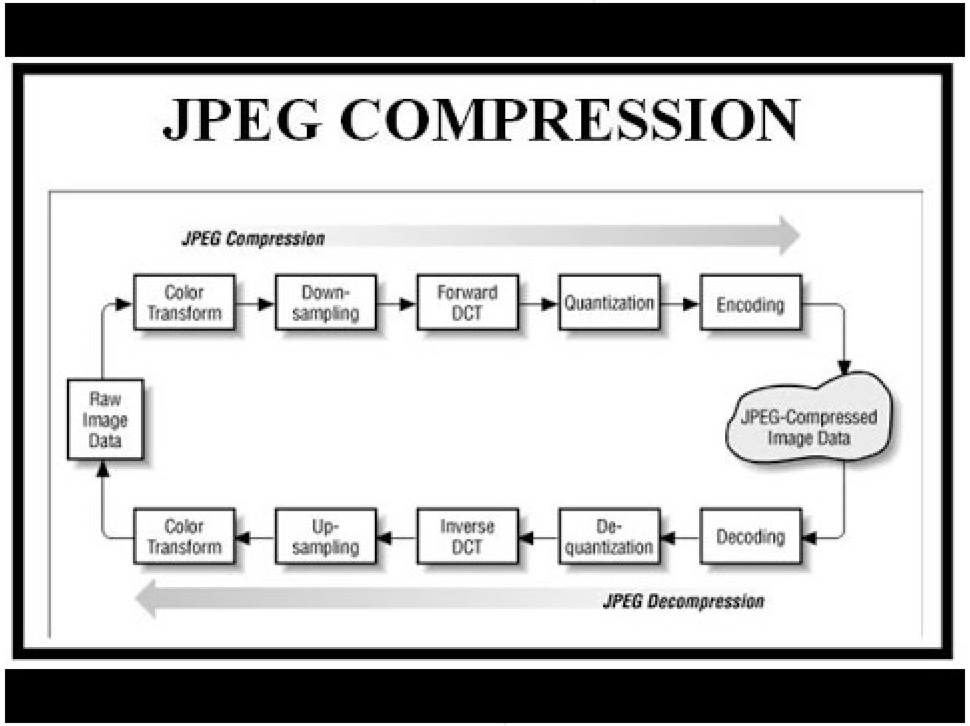



  - HSL

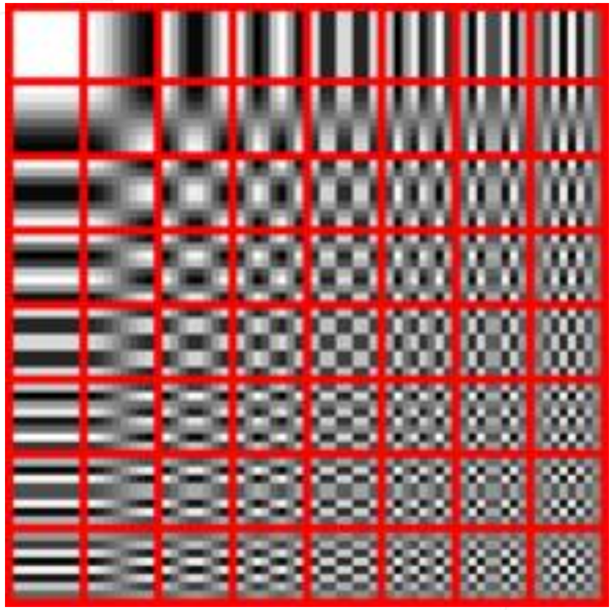



  - CMYK

- **Compression methods for image**
  - JPG, PNG, GIF, Webm

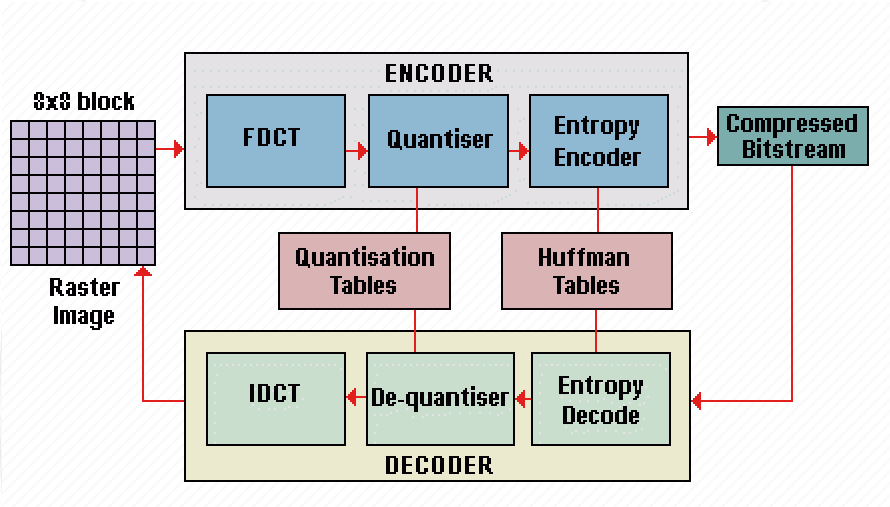



For JPG: discrete cosine transform



The DCT transforms an 8×8 block of input values to a linear combination of these 64 patterns. The patterns are referred to as the two-dimensional DCT basis functions, and the output values are referred to as transform coefficients.

- **Compression method for video**
  - H.261, H.262, H.263, H.264, H.265, AV1, WMV

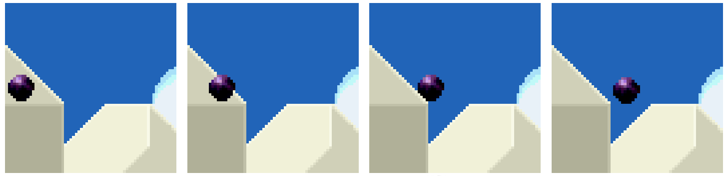


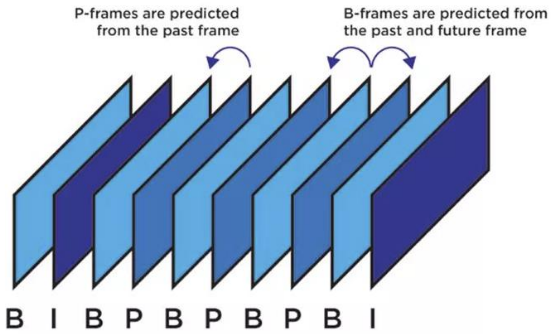

Example: Encoder decoder structure

# Image & video Representation

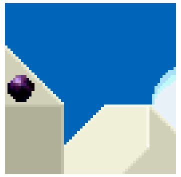
- **Frame types of video**

Input:



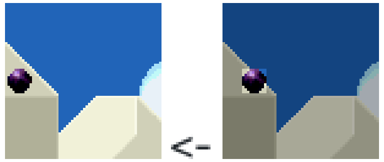



- **I Frame (intra, keyframe)**



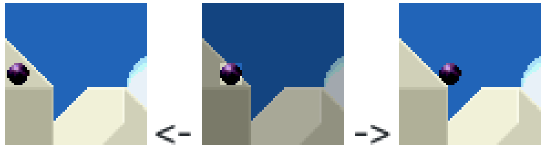
An I-frame (reference, keyframe, intra) is a self-contained frame. It doesn't rely on anything to be rendered, an I-frame looks similar to a static photo.

- **P Frame (predicted)**




A P-frame takes advantage of the fact that almost always the current picture can be rendered using the previous frame.

- **B Frame (bi-predictive)**




B-frame refers the past and future frames to provide even a better compression

# Outline

- **History of Digital Image Processing**

**1960s**: Improvements in computing technology and the onset of the **space race** led to a surge of work in digital image processing

- **1964**: Improve the quality of images of moon
- Such techniques were used in Apollo landings

Image Enhancement

A picture of the moon taken by the Ranger 7 probe minutes before landing

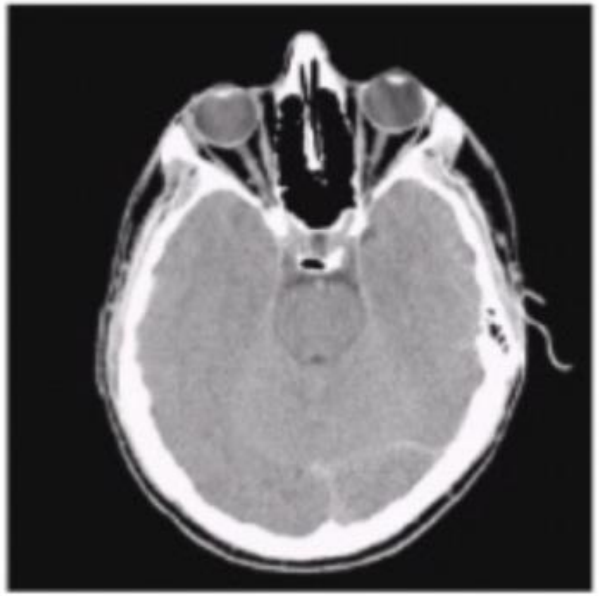
# Image Processing

- **History of Digital Image Processing**

**1970s**: Digital Image processing begins to be used in medical applications

- **1979:** Sir Godfrey & Prof. Allan share the Nobel Prize in medicine for the tomography.

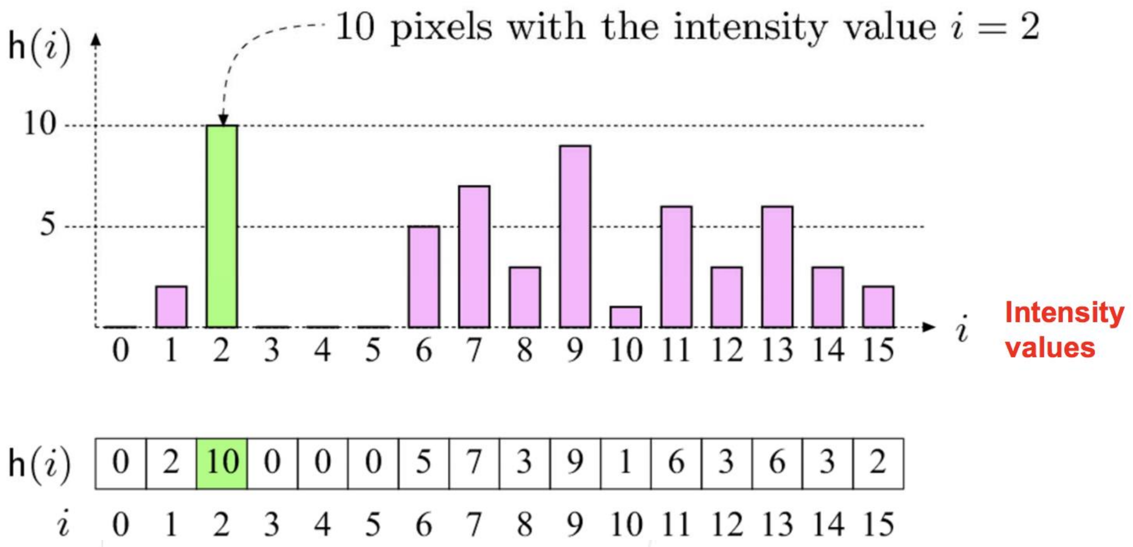
Image Restoration



Typical head slice CAT image

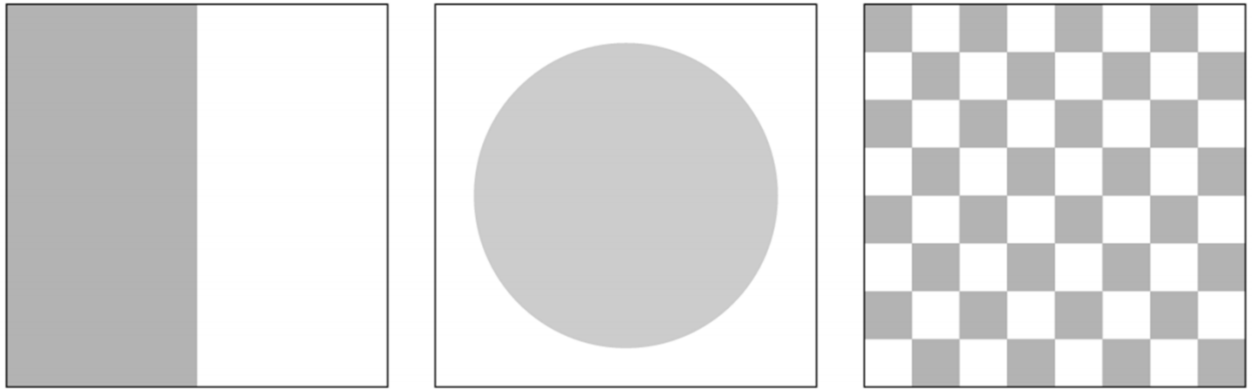
- **Histograms**
  - Histograms plots how many times(frequency) each intensity value in image occurs
  - **Example:**
    - Image (left) has 256 distinct gray levels (8 bits)
    - Histogram (right) shows frequency (how many times) each gray level occurs

- **Histograms**




- Histograms: only statistical information
- No indication of location of pixels

- **Histograms**
  - Different images can have same histogram
  - 3 images below have same histogram



  - Half of pixels are gray, half are white
    - Same histogram = same statistics
    - Distribution of intensities could be different

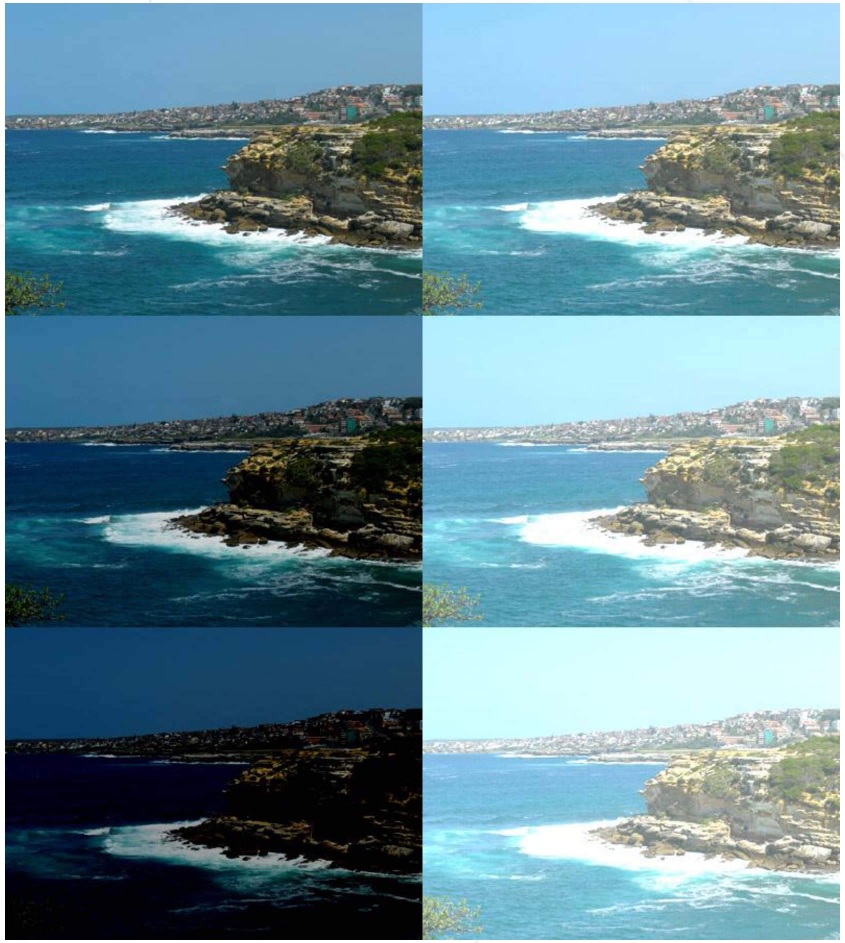
- **Brightness**

  - Brightness of a grayscale image is the average intensity of all pixels in image

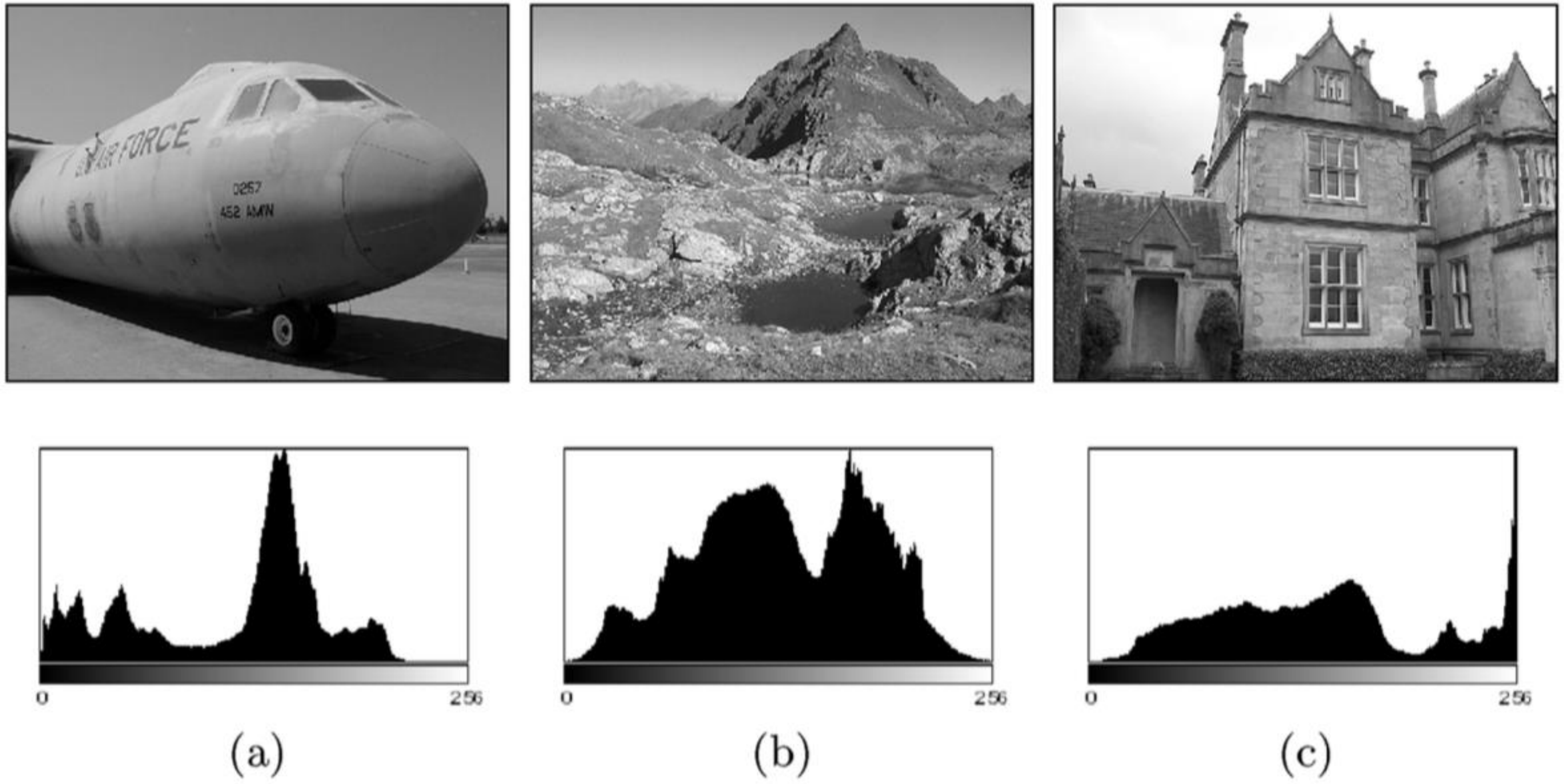
$$B(I) = \frac{1}{wh} \sum_{v=1}^{h} \sum_{u=1}^{w} I(u, v)$$

1. Sum up all pixel intensities

2. Divide by total number of pixels



- **Brightness and Histogram**




Image

Histogram

(a) Underexposed

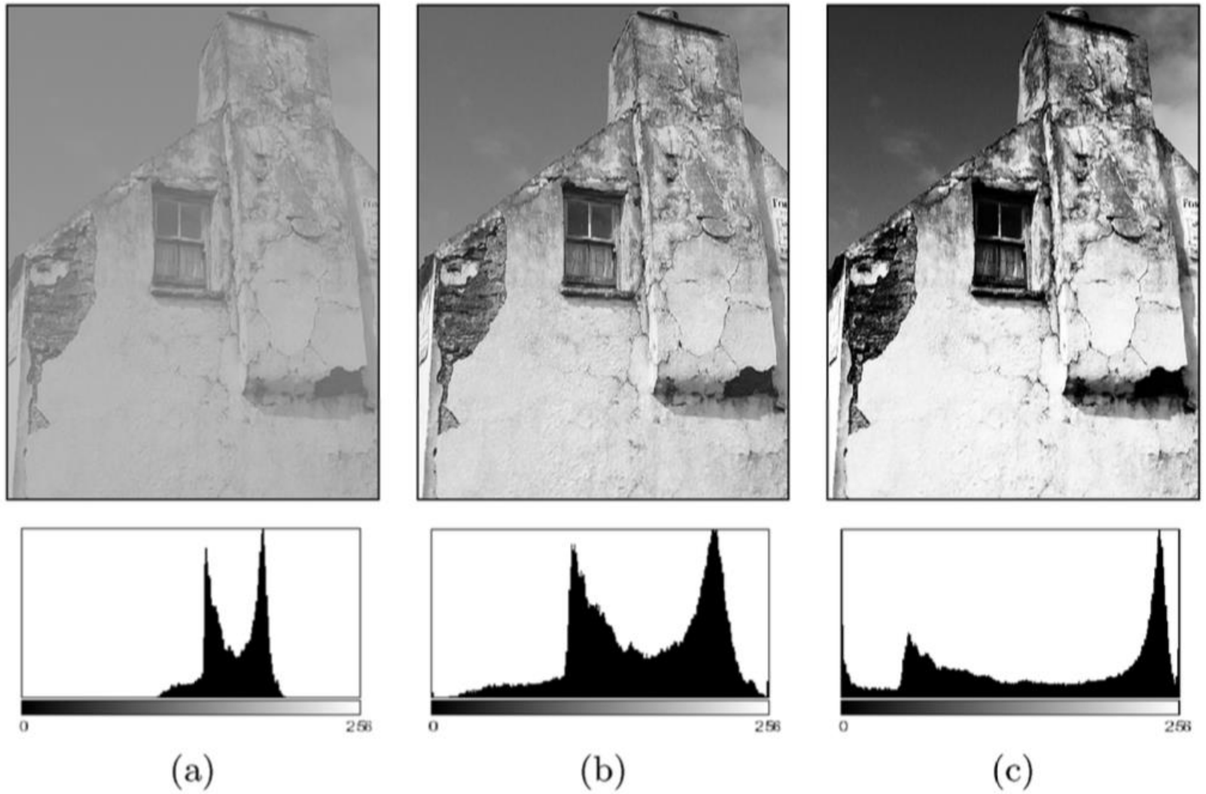
(b) Properly Exposed

(c) Overexposed

- **Image Contrast**
  - The contrast of a grayscale image indicates how easily objects in the image can be distinguished
    - **High contrast:** many distinct intensity values
    - **Low contrast:** image uses few intensity values
  - Many different equations for contrast exist

$$\text{Contrast} = \frac{\text{Change in Luminance}}{\text{Average Luminance}}$$

- **Contrast and Histogram**




Image

Histogram

Low contrast

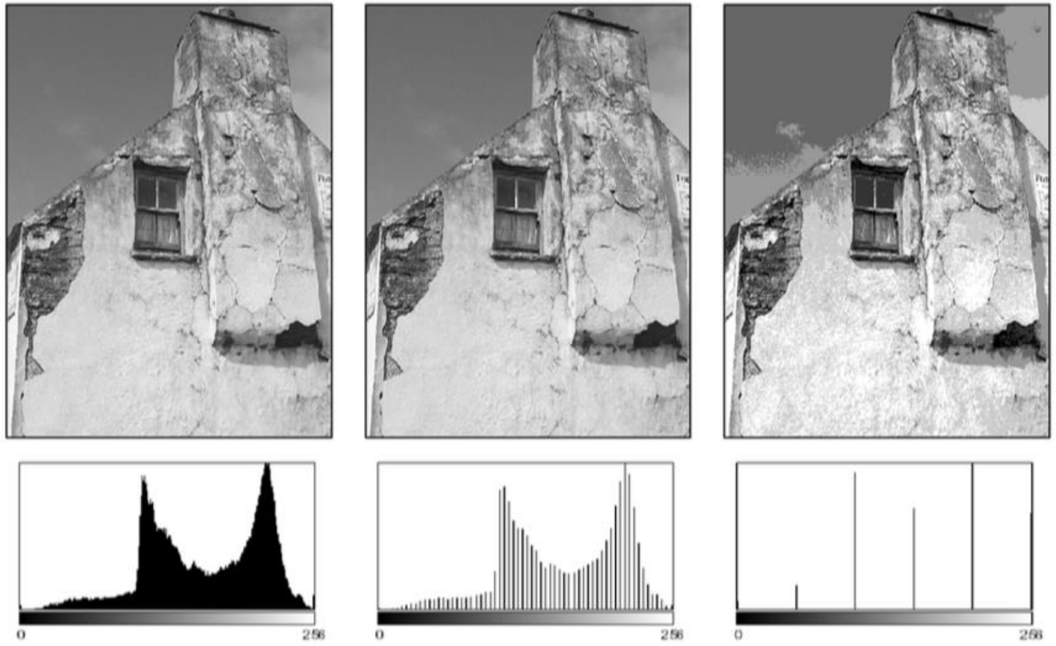
Normal contrast

High contrast

# Image Processing

- ## Dynamic Range and Histogram

- **Dynamic Range:** Number of distinct pixels in image




(a) **High Dynamic Range**

(b) **Low Dynamic Range (64 intensities)**

(c) **Extremely low Dynamic Range (6 intensity values)**

- **Image Enhancement - intensity transformation**

  - **Image negatives**
  - Transform function T : g(x, y) = L - f(x, y),

    where L is the max intensity.

```python
import cv2
import numpy as np
# Load the image
img = cv2.imread('D:/downloads/forest.jpg')
# Check the datatype of the image
print(img.dtype)
# Subtract the img from max value(calculated from dtype)
img_neg = 255 - img
# Show the image
cv2.imshow('negative',img_neg)
cv2.waitKey(0)
```

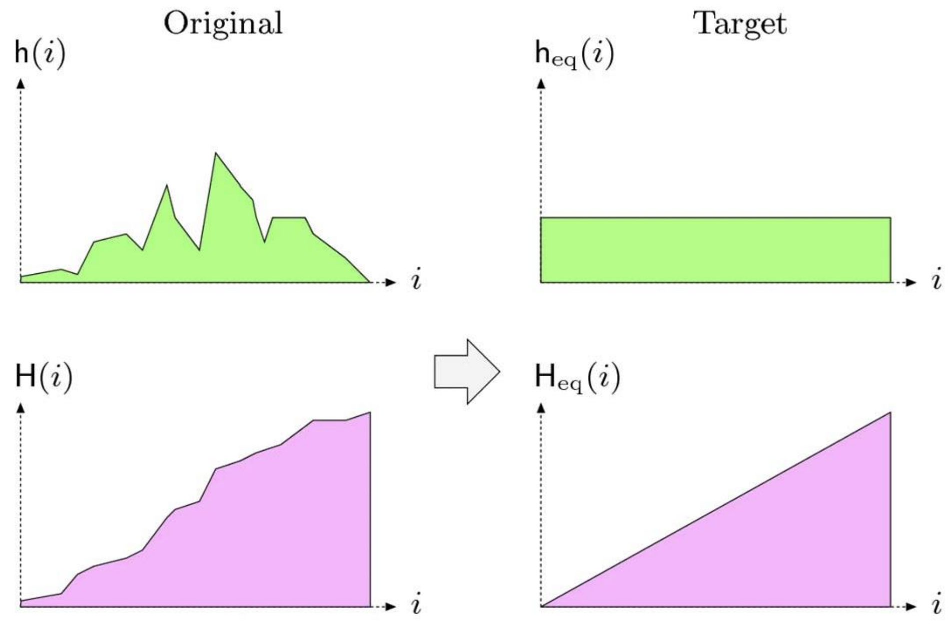
Original

Negative

- **Image Enhancement - Histogram equalization**
  - Apply a point operation that changes histogram of modified image into **uniform distribution**

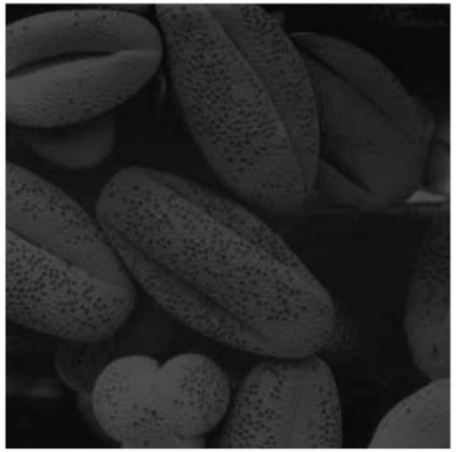
Histogram

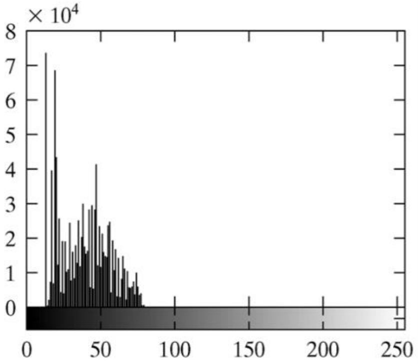
Cumulative Histogram

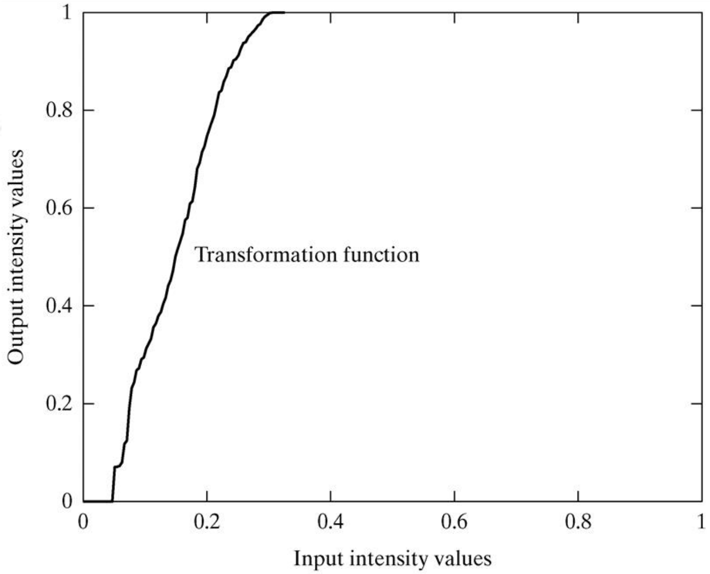


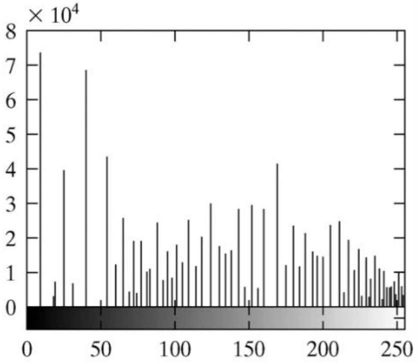

```
img = cv2.imread('test.jpg',0)
equ = cv2.equalizeHist(img)
res = np.hstack((img,equ)) #stacking images side-by-side
cv2.imwrite('output.png',res)
```

- **Image Enhancement - Histogram equalization**





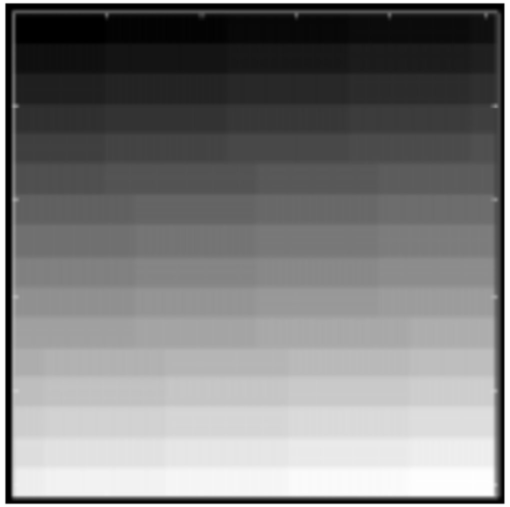



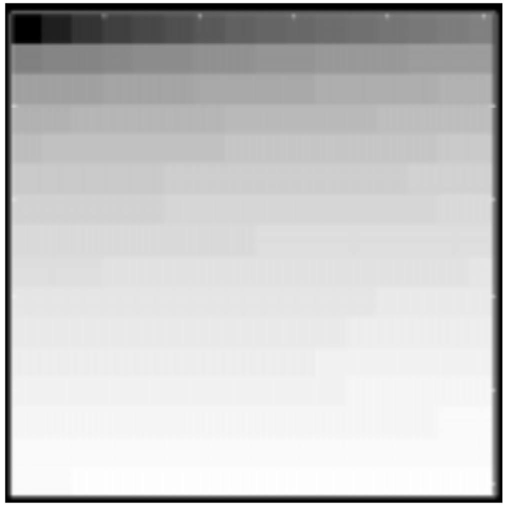
- **Image Enhancement - Compression of dynamic range**

$s = c \log(1+|r|)$

- where c is a scaling constant, and the logarithm function performs the desired compression.



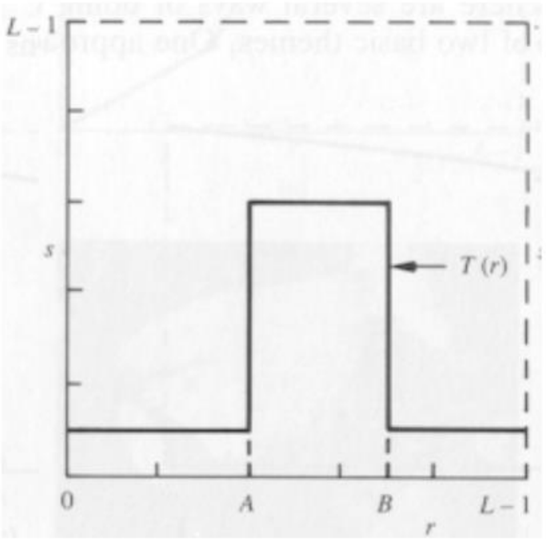
Original

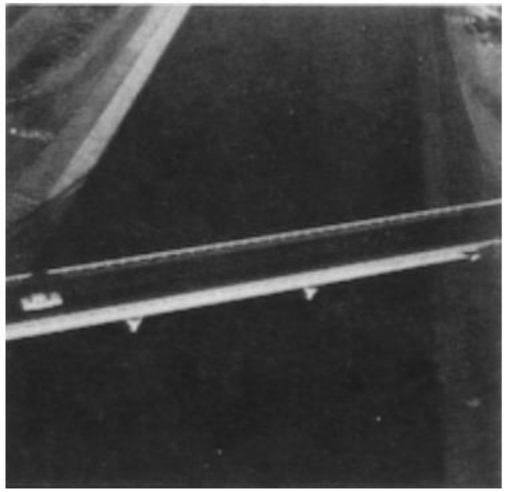


Processed output

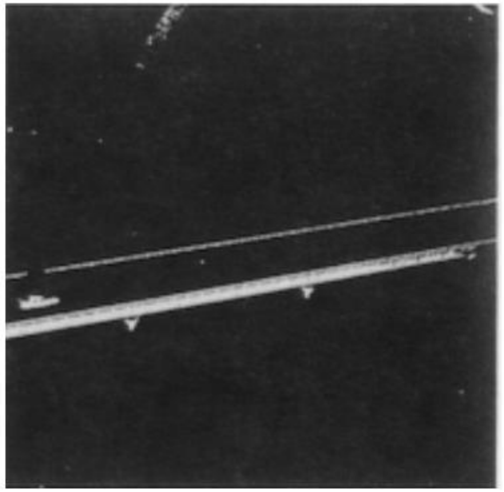
## Image Enhancement - Gray-level slicing

- A function that highlights a range [A,B] of transformation intensities while diminishing all others to a constant.



(b)



(c)

Fig 1. (a) Transfer function, (b) Original image, (c) Processing output.

# Image Enhancement - Spatial Filtering

1. Low pass filtering

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

2. Median filtering

replacing each point with the median of neighboring points.

3. Sharpening Filter

$$\frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Original with (a) spike noise (b) white noise

Median filtering output

Low-pass filtering output

- **Image Enhancement in the frequency domain**

Spatial domain: $g(x,y)=f(x,y)*h(x,y)$

$\Updownarrow$

Frequency domain: $G(w_1,w_2)=F(w_1,w_2)H(w_1,w_2)$

- Lowpass filtering

$$H(u,v)=\begin{cases}1 & \text{if } D(u,v)\le D_o\\ 0 & \text{else}\end{cases}$$

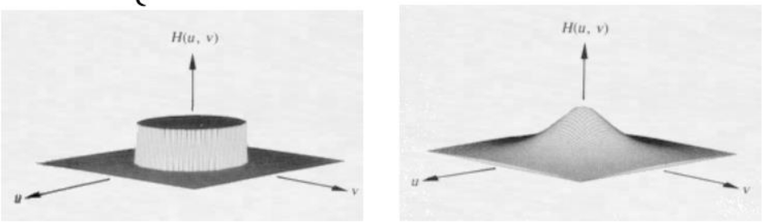



Fig 6. (a) Ideal LPF; (b) Butterworth LPF.

- Highpass filtering

$$H(u,v)=\begin{cases}0 & \text{if } D(u,v)\le D_o\\ 1 & \text{else}\end{cases}$$

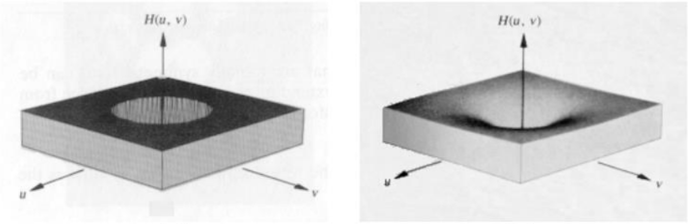
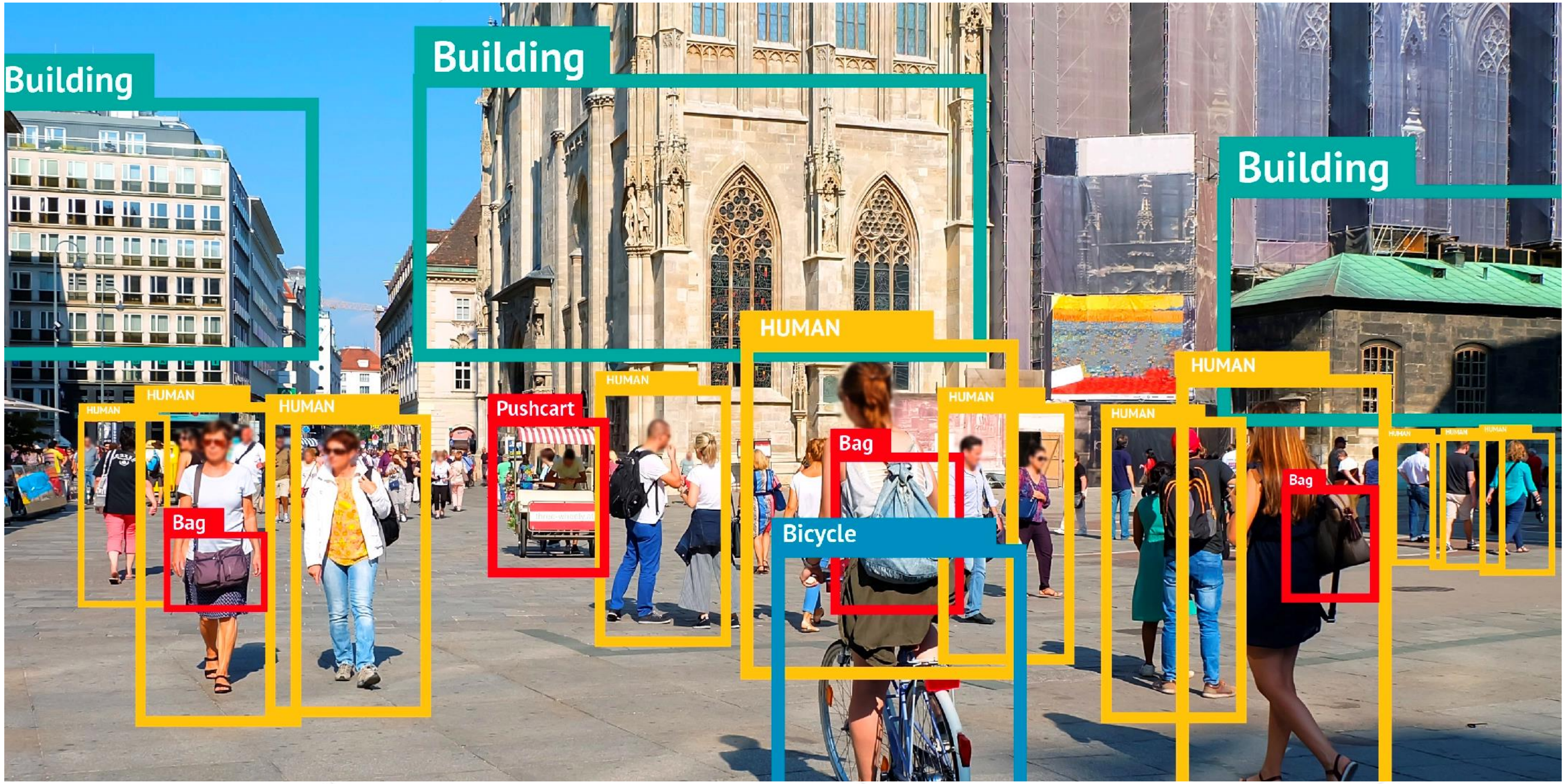
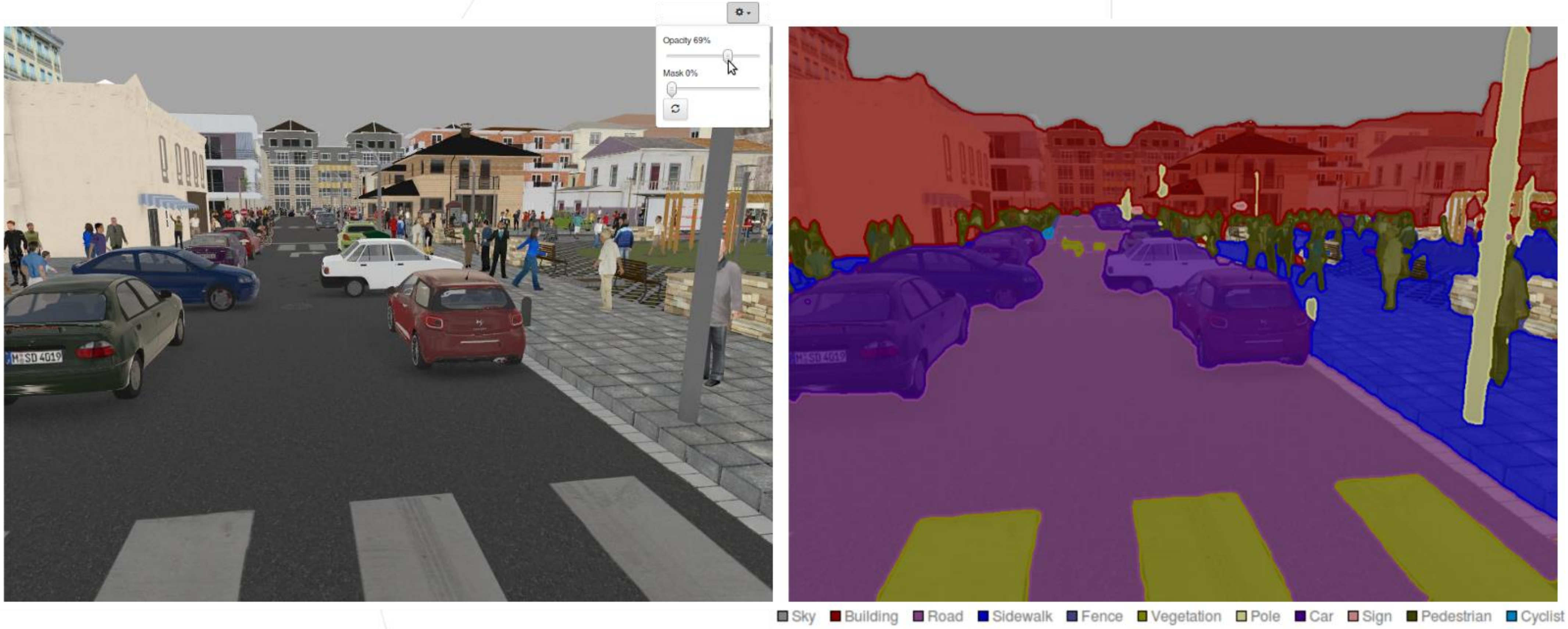



Fig 7. (a) Ideal HPF; (b) Butterworth HPF.

- **Image Detection**



- **Image Segmentation**



Tsinghua University

# Outline

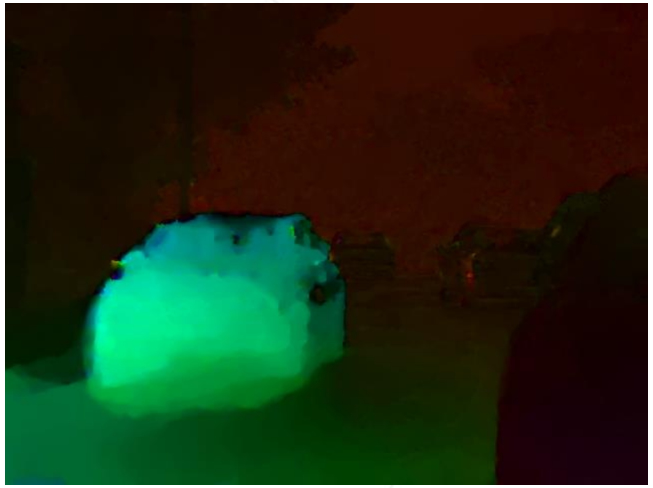
| | |
|---|---|
| Part 1 | Image and video representation |
| Part 2 | Image processing |
| Part 3 | Video processing |

- **Optical Flow**

Optical flow is the pattern of apparent  motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene.

T = t

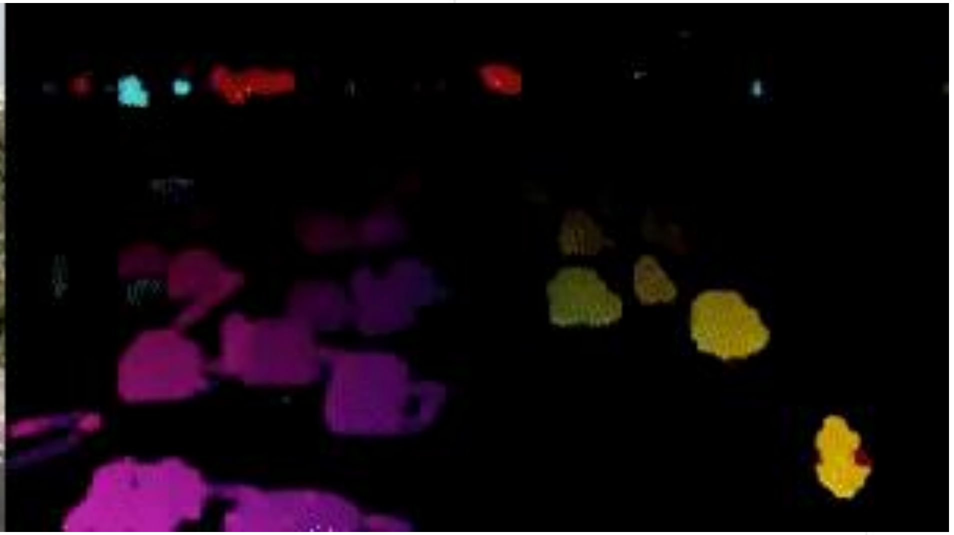
T = t + 1



Optical flow

- **Two types of Optical Flow**

Sparse



Dense

*Gif by: https://nanonets.com/blog/optical-flow/*

- **Optical Flow demo**


*Gif by: https://gfycat.com/fr/wetcreepygecko*

- **Optical Flow Estimation**

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

- Assuming the movement is small

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \text{higher-order terms}$$

- By truncating the higher order terms, a linearization, it follows that

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0 \qquad \frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0$$

- Thus

$$I_x V_x + I_y V_y = -I_t$$

- This is an equation in two unknowns and cannot be solved as such. This is known as the aperture problem of the optical flow algorithms
- To find the optical flow another set of equations is needed, given by some additional constraint. All optical flow methods introduce additional conditions for estimating

- **Lucas–Kanade method (Sparse, Local)**
  - It assumes that the flow is essentially constant in a local neighborhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighborhood, by the least squares criterion

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

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

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

  - Since it is a purely local method, it cannot provide flow information in the interior of uniform regions of the image.

- **Horn–Schunck method (Dense, Global)**

The Horn-Schunck algorithm assumes smoothness in the flow over the whole image. Thus, it tries to minimize distortions in flow and prefers solutions which show more smoothness.

Let the image be p = (x,y) and the underlying flow field be w(p) = (u(p),v(p), 1), where u(p) and v(p) are the horizontal and vertical components of the flow field, respectively.

$$\mathrm{E}(\mathrm{u},\mathrm{v}) = \int |\mathrm{I}_2(\mathbf{p}+\mathbf{w}) - \mathrm{I}_1(\mathbf{p})|^2 + \lambda(|\nabla u|^2 + |\nabla v|^2)d\boldsymbol{p}$$

To solve Eq. (1), we use an iterative flow framework. It assumes that an estimate of the flow field is w, and one needs to estimate the best increment dw(dw=(du,dv)), to update w. The objective function in Eq. (1) is then changed to

$$\mathrm{E}(\mathrm{du},\mathrm{dv}) = \int |\mathrm{I}_2(\mathbf{p}+\mathbf{w}+\mathbf{dw}) - \mathrm{I}_1(\mathbf{p})|^2 + \lambda(|\nabla(u+du)|^2 + |\nabla(v+dv)|^2)d\boldsymbol{p}$$

The main idea to solve the above equation is to find dU,dV so that the gradient

$$\left[\frac{\partial E}{\partial dU}; \frac{\partial E}{\partial dV}\right] = 0$$

- **Horn–Schunck method**

We can derive

$$\frac{\partial E}{\partial dV} = 2((\boldsymbol{I}_y^2 + \lambda \boldsymbol{L})dV + \boldsymbol{I}_x\boldsymbol{I}_y dU + \boldsymbol{I}_y I_z + \lambda \boldsymbol{L}V)$$

where L is a Laplacian filter defined as

$$I_z(\mathbf{p}) = I_2(\mathbf{p} + \mathbf{w}) - I_1(\mathbf{p})$$

$$\boldsymbol{L} = \boldsymbol{D}_x^T \boldsymbol{D}_x + \boldsymbol{D}_y^T \boldsymbol{D}_y \qquad I_x(\mathbf{p}) = \frac{\partial}{\partial x} I_2(\mathbf{p} + \mathbf{w})$$

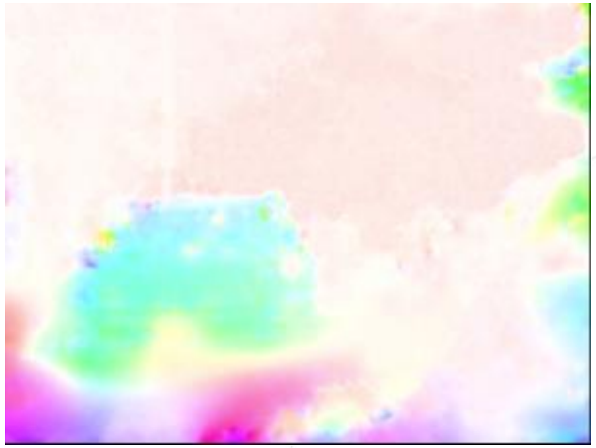
$$I_y(\mathbf{p}) = \frac{\partial}{\partial y} I_2(\mathbf{p} + \mathbf{w})$$

The term of dU in gradient is derived similarly. Therefore, solving the gradient equation can be performed in the following linear system

$$\begin{bmatrix} \boldsymbol{I}_x^2 + \lambda \boldsymbol{L} & \boldsymbol{I}_x \boldsymbol{I}_y \\ \boldsymbol{I}_x \boldsymbol{I}_y & \boldsymbol{I}_y^2 + \lambda \boldsymbol{L} \end{bmatrix} \begin{bmatrix} dU \\ dV \end{bmatrix} = - \begin{bmatrix} \boldsymbol{I}_x I_z + \lambda \boldsymbol{L}U \\ \boldsymbol{I}_y I_z + \lambda \boldsymbol{L}V \end{bmatrix}$$

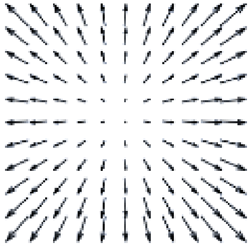
- **Horn–Schunck method**

Input two frames



Dense optical flow

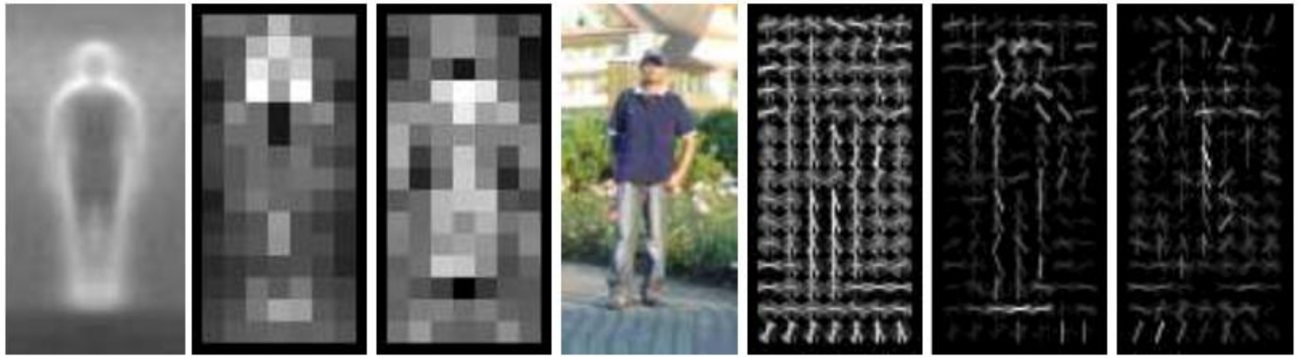
Wrapped frame



Flow Visualization

- **Video Descriptors**

HOG: Histogram of oriented spatial grad



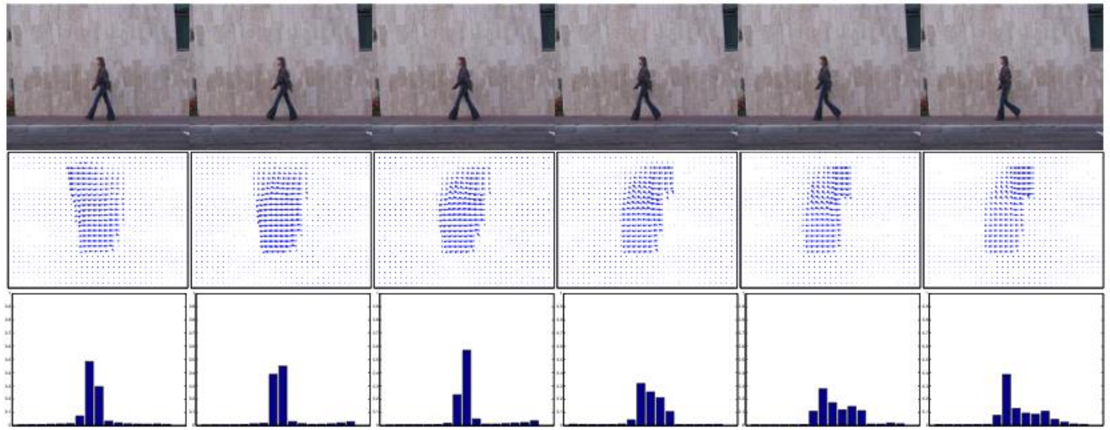
(a) (b) (c) (d) (e) (f) (g)

(a) The average gradient image over the training examples.
(b) Each ‘pixel’ shows the maximum positive SVM weight in the block centred on the pixel.
(c) Likewise for the negative SVM weights.
(d) A test image.
(e) It’s computed R-HOG descriptor.
(f,g) The R-HOG descriptor weighted by respectively the positive and the negative SVM weights.

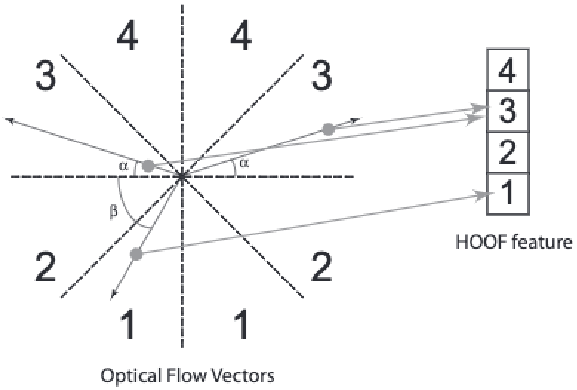
*Navneet Dalal, Bill Triggs. Histograms of Oriented Gradients for Human Detection. International Conference on Computer Vision & Pattern Recognition (CVPR ’05), Jun 2005, San Diego, United States. pp.886–893, 10.1109/CVPR.2005.177. inria-00548512*

- **Video Descriptors**

HOF: Histogram of oriented optical flow



Optical flows and HOF feature trajectories




Histogram formation with four bins, B= 4

*Chaudhry R, Ravichandran A, Hager G, et al. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions[C]//2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009: 1932-1939.*

- ## Video Descriptors

MBH: Motion Boundary Histograms

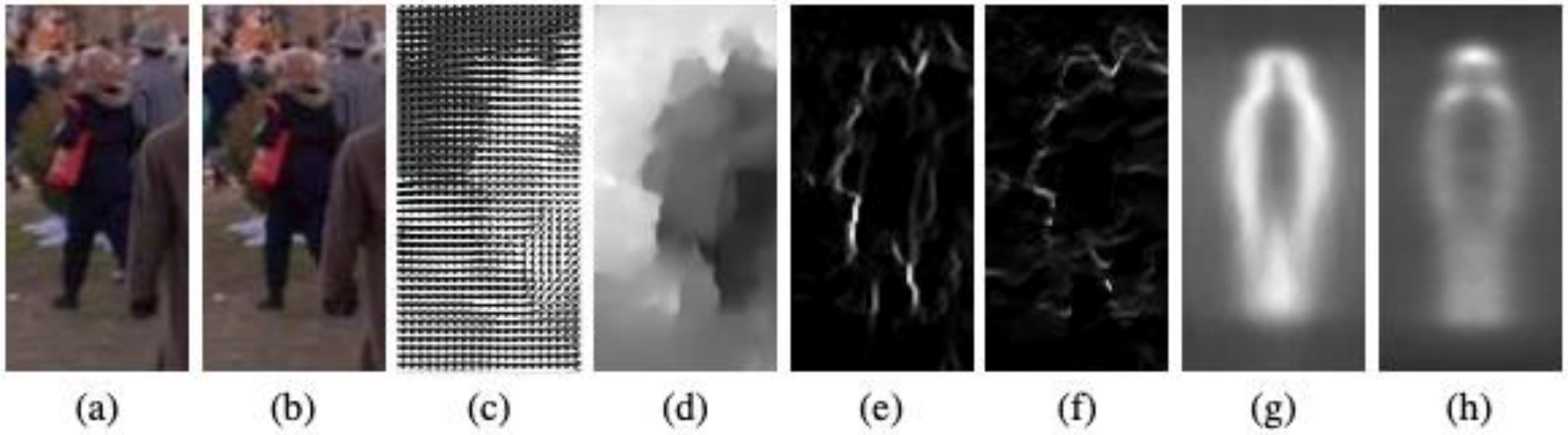
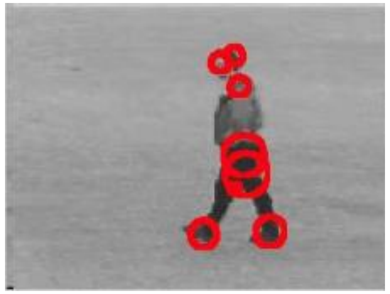



Illustration of the MBH descriptor.
(a,b) Reference images at time $t$ and $t+1$.
(c,d) Computed optical flow, and flow magnitude showing motion boundaries. (e,f) Gradient magnitude of flow field $\mathcal{I}^x, \mathcal{I}^y$ for image pair (a,b). (g,h) Average MBH descriptor over all training images for flow field $\mathcal{I}^x, \mathcal{I}^y$.

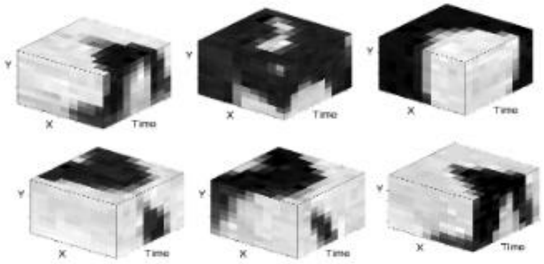
*Dalal N, Triggs B, Schmid C. Human detection using oriented histograms of flow and appearance[C]//European conference on computer vision. Springer, Berlin, Heidelberg, 2006: 428-441.*

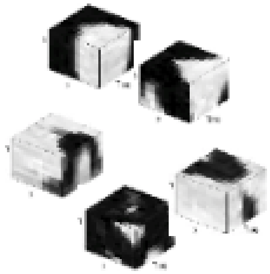
# Video Processing

- **Traditional Action classification**
  - Bag of space-time features + SVM



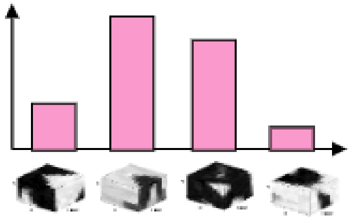
Collection of space-time patches





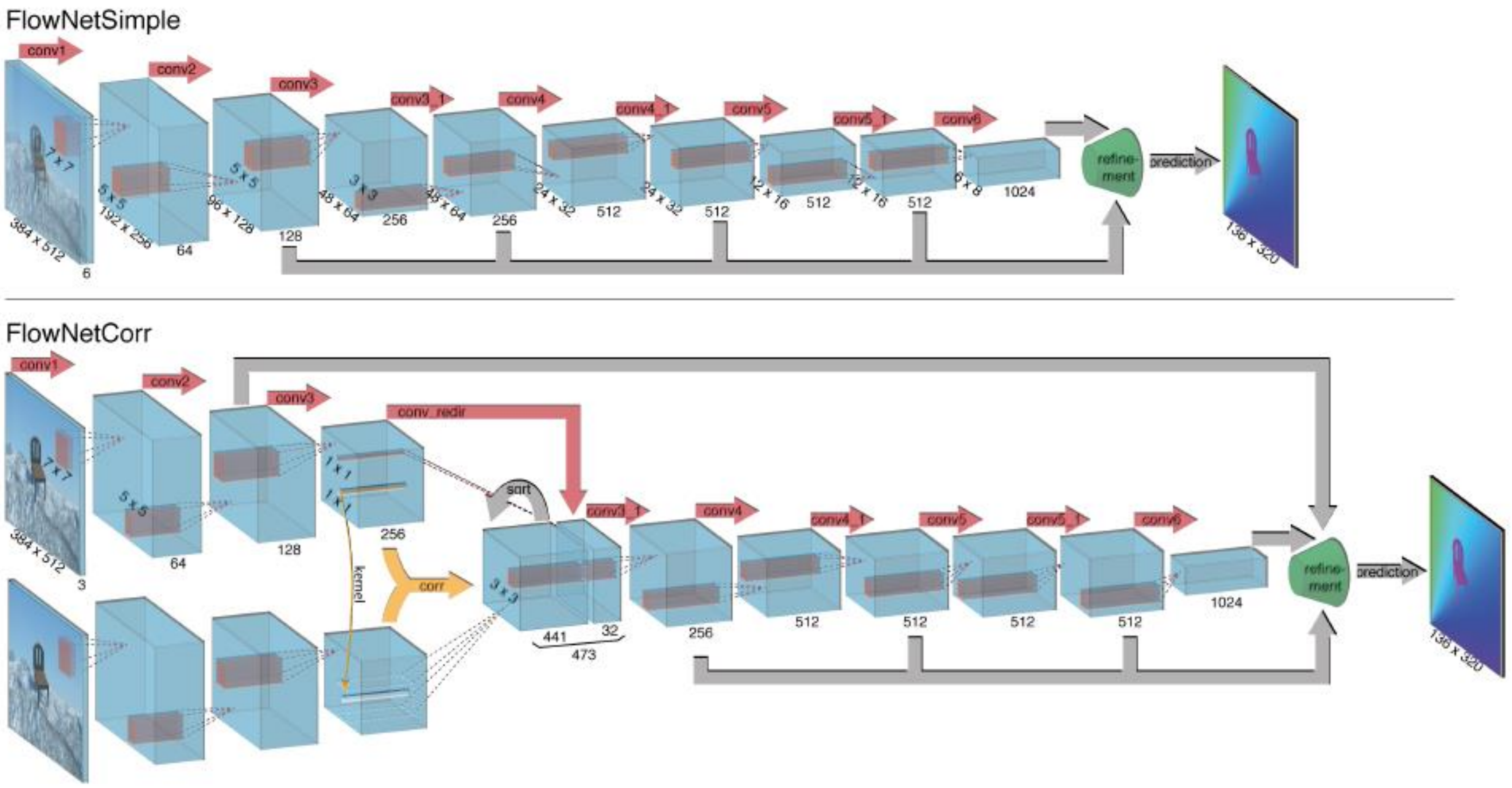
HOG & HOF patch& Others descriptors

Histogram of visual words



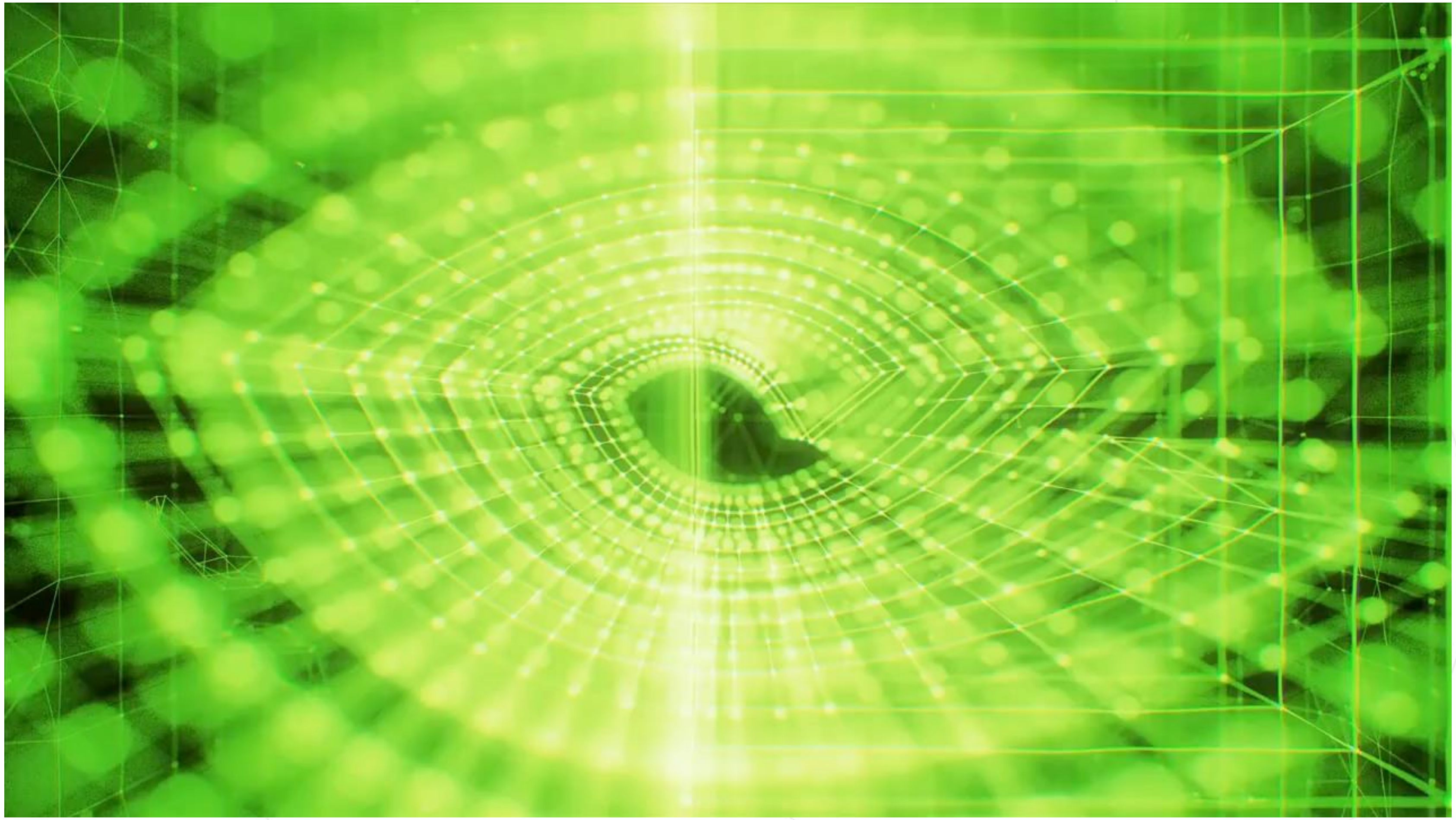
SVM Classifier

- **Optical flow estimation via deep networks**




Fischer et al.: FlowNet: Learning Optical Flow with Convolutional Networks, ICCV 2015.

- **Video Interpolation**



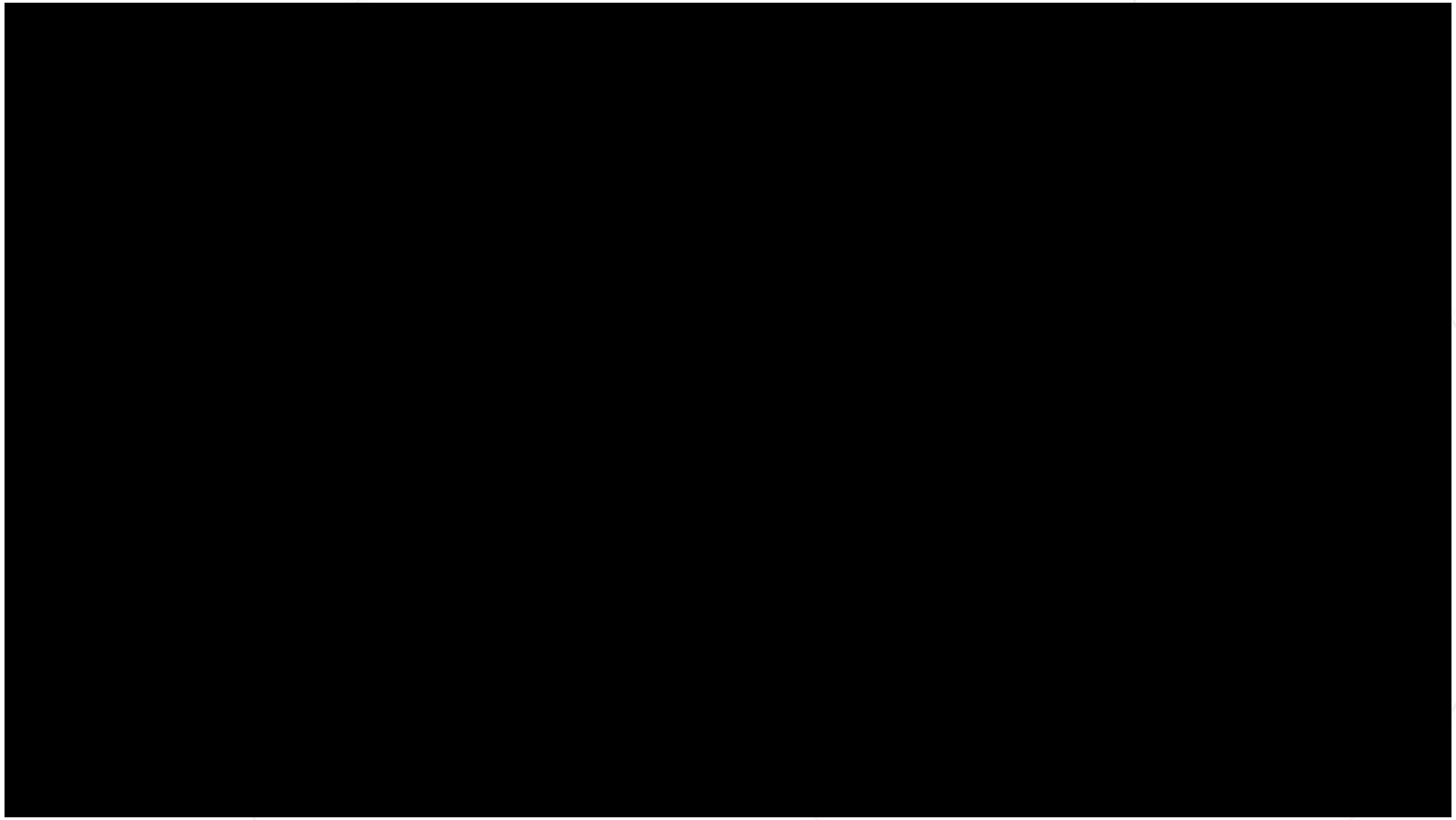
http://jianghz.me/projects/superslomo/

- **Video Stabilization**

- **Video Denosing**

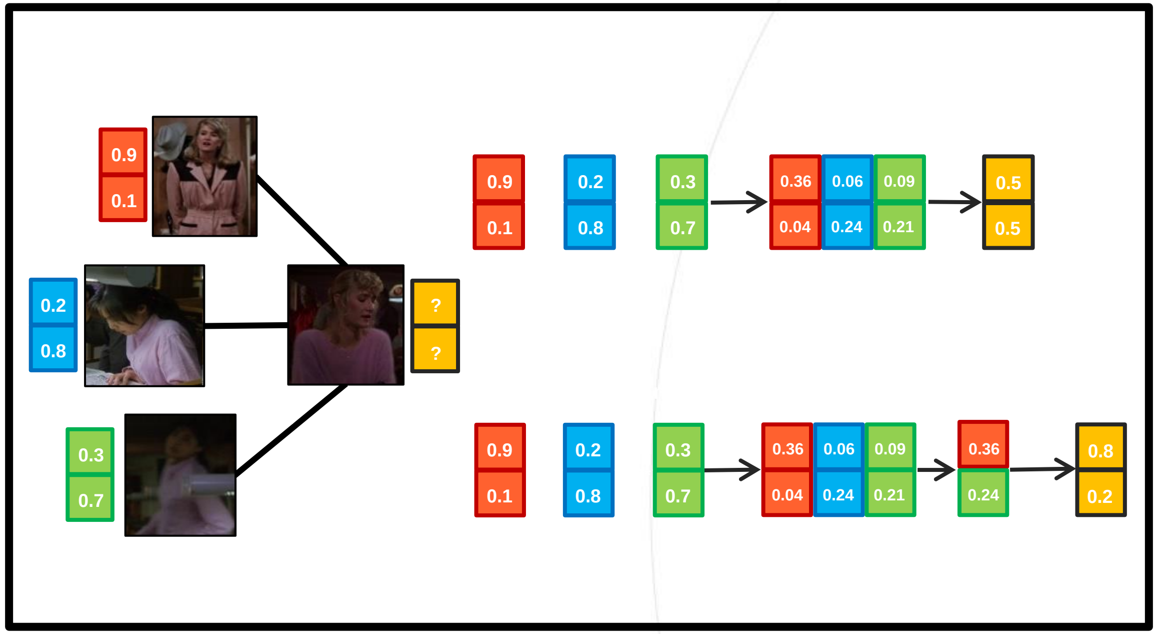
# Video Processing

- **Video Super-Resolution**



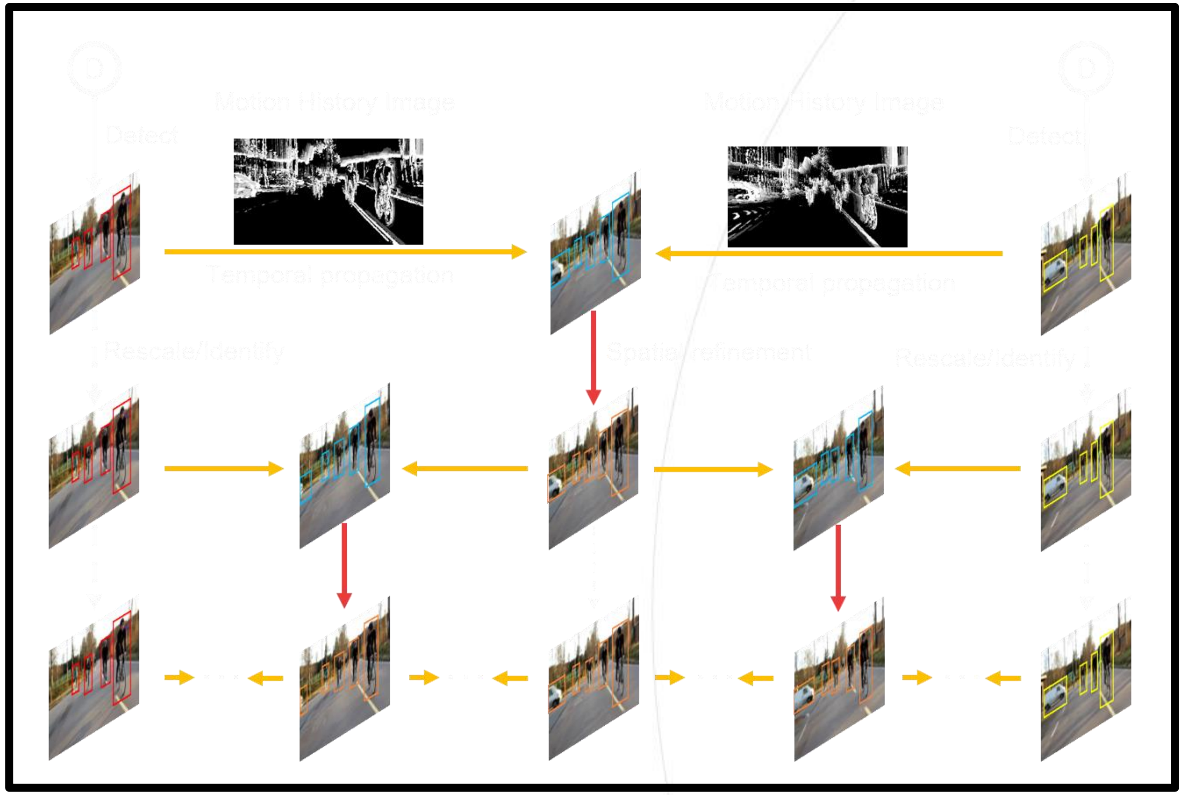
# Video Processing

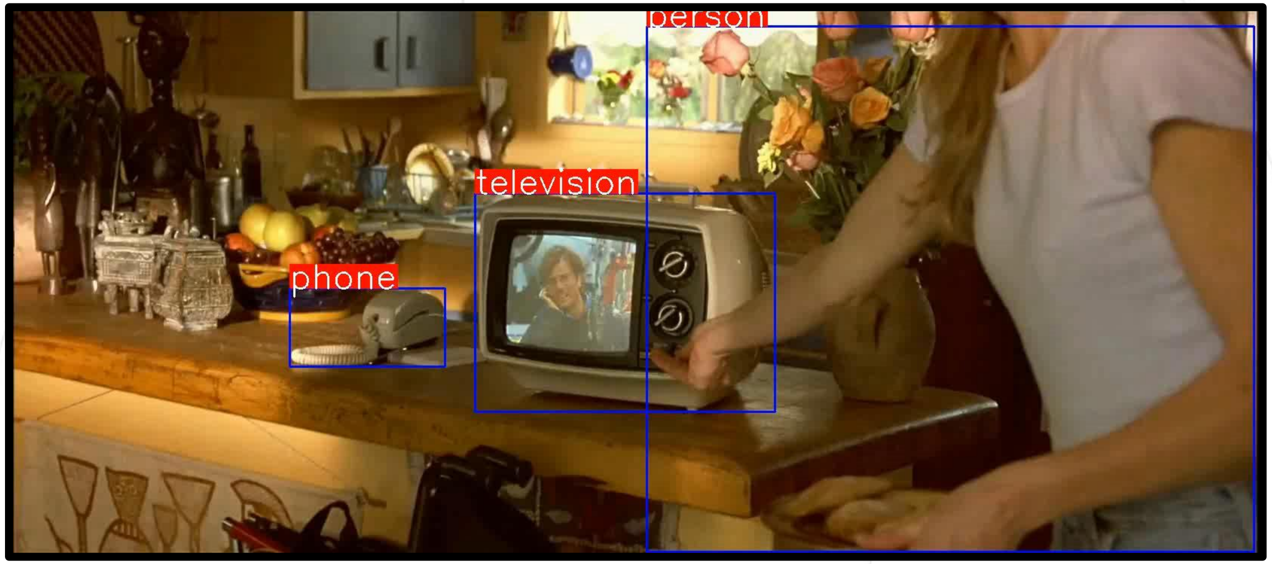
- **Video Understanding - Human**




*Q. Huang, W. Liu, D. Lin. Person Search in Videos with One Portrait Through Visual and Temporal Links. ECCV 2018*

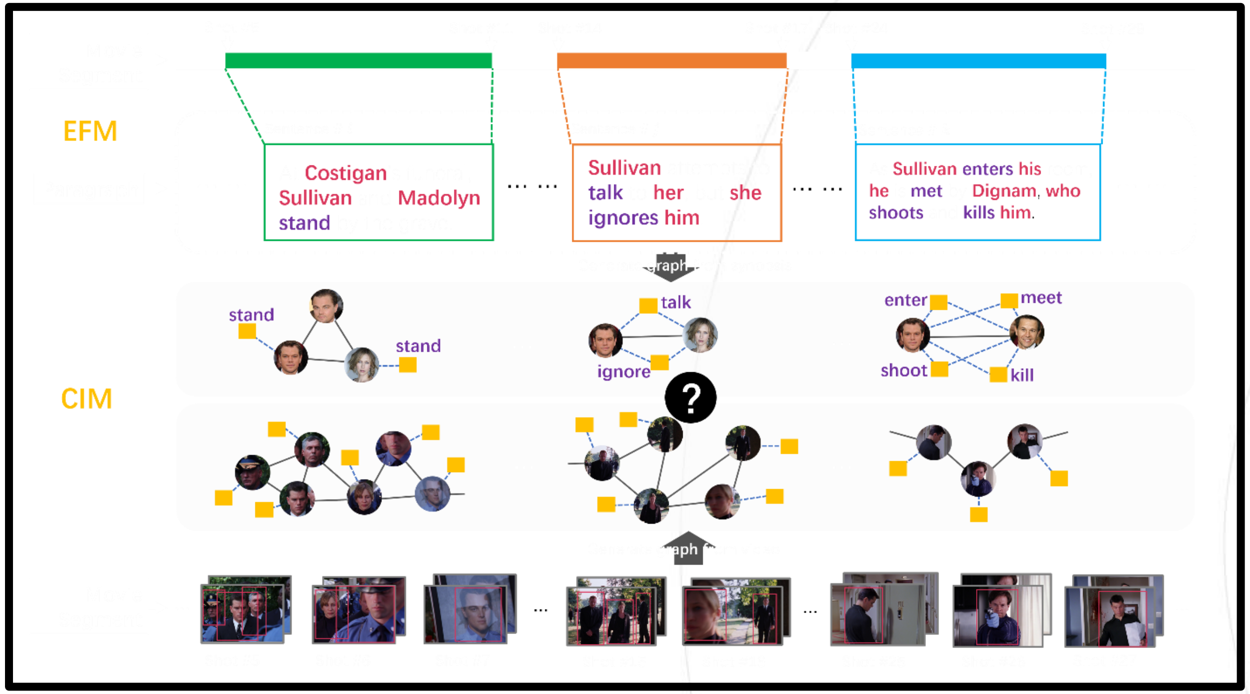
- **Video Understanding - Object**






*K. Chen et al. Optimizing Video Object Detection via a Scale-Time Lattice. CVPR 2018.*

- **Video Understanding - Context**




*Y. Xiong, Q. Huang, L. Guo, H. Zhou, B. Zhou, D. Lin. A Graph-based Framework to Bridge Movies and Synopses. ICCV 2019.*