Computer Vision Course — A.A. 2021/2022

Lab 2: Motion Detection

Nicola Garau
nicola.garau@unitn.it

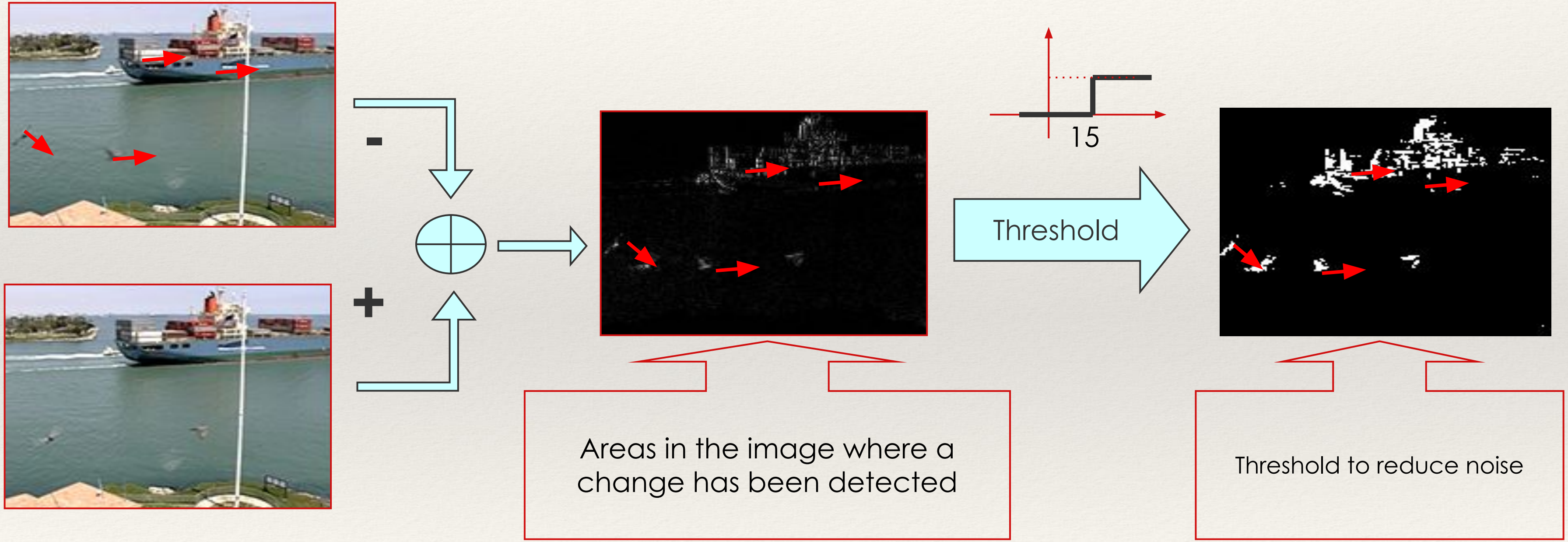
Room change

- Next time the lecture will be in A110
- 11.30-13.30 as usual

What's up today?

- Frame differencing
- Background Subtraction
- Adaptive Background Subtraction
- Adaptive Background Subtraction: Mixture of Gaussians

Frame differencing

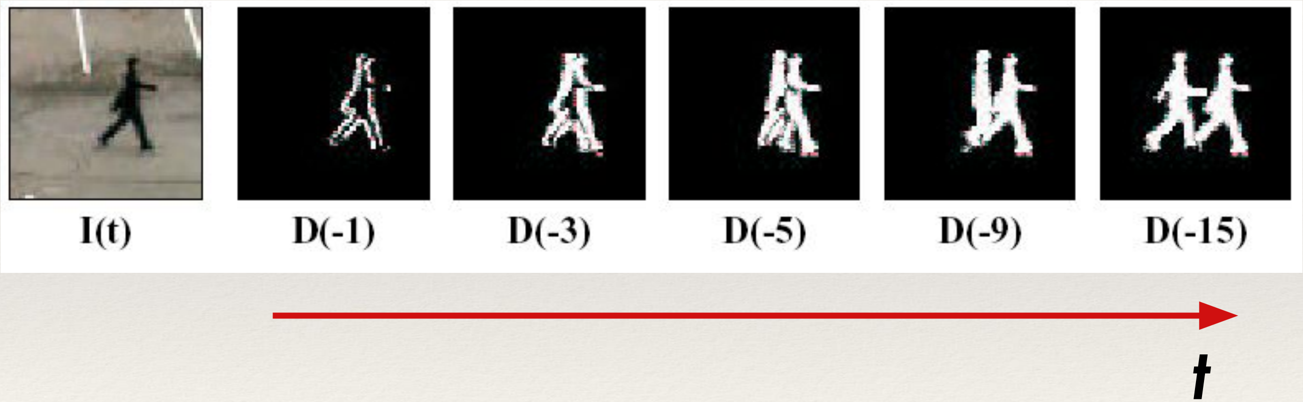


Areas in the image where a change has been detected

Threshold to reduce noise

Frame Differencing: Time Scaling

$$D(N) = \left\| I(t) - I(t+N) \right\|$$

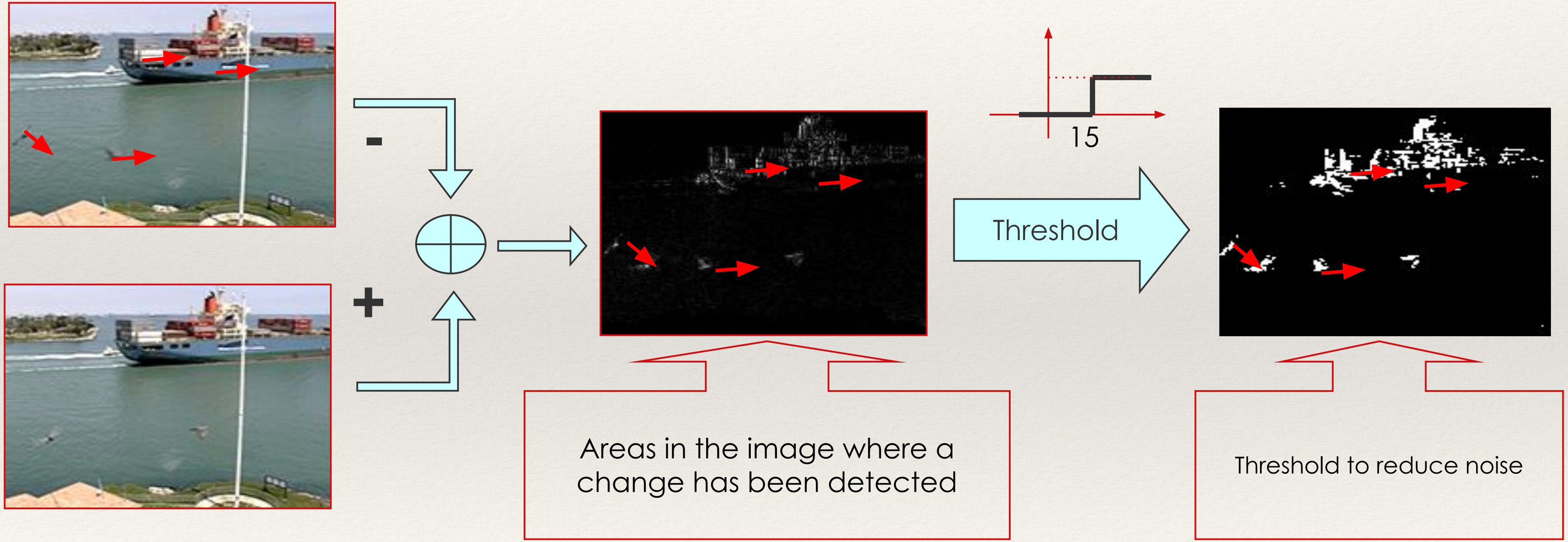


Exercise: frame differencing

- Initialise a new project
- Open a video
- Convert frames to grayscale
  - `frame_gray = cv2.cvtColor(frame_color, cv2.COLOR_RGB2GRAY)`
- Use a list to store frames!
  - `frames = [ ]`
  - 💡 How to append frames to array: `frame.append(frame_gray)`
- Implement the function
  - `result = cv2.absdiff(I(t),I(t+N))`
  - 💡 Hint: **`t+N`** can be thought as **`t-N`**

$$D(N) = \left\| I(t) - I(t+N) \right\|$$

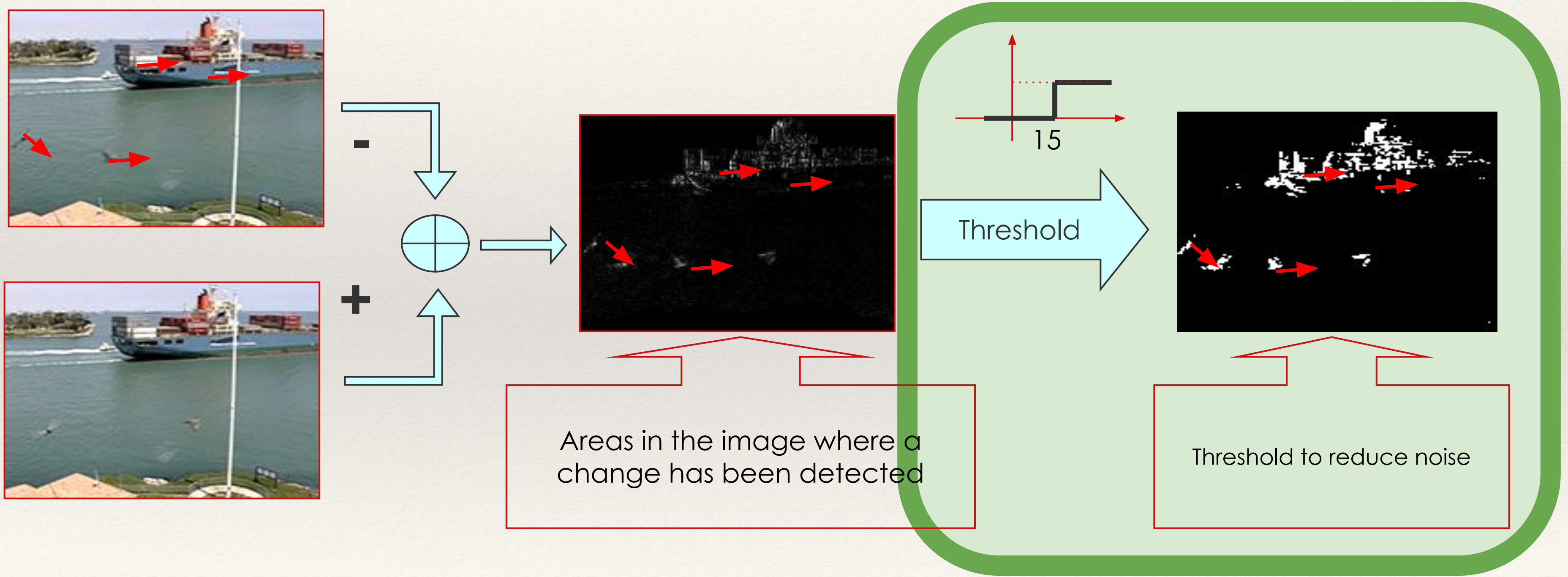
What’s missing in the implementation?



Areas in the image where a change has been detected

Threshold to reduce noise

Apply thresholding on the mask



Areas in the image where a change has been detected

Threshold to reduce noise

```
cv2.threshold(diff, THRESH, MAXVAL, TYPE)
```

Adaptive Background Subtraction

- Use a parameter **α** *(learning rate)* to weight the contributions
  - $B_t = \alpha I_t + (1 - \alpha) B_{t-1}$
    - $\alpha = 0$ □ bg sub, no update
    - $\alpha = 1$ □ frame differencing

Material

- Inside the Virtual Machine (or in your programming environment)
- Go to this link and download the file
- https://github.com/nick1392/MOG/archive/master.zip

Mixture of Gaussians

$$P(x_t) = \sum_{i=1}^{K} \omega_{i,t}\eta(x_t,\mu_{i,t},\Sigma_{i,t})$$

- $\omega_{i,t}$ is the weight for the current Gaussian
- Select *K*
- Rank the Gaussians on the basis of
  - Peak amplitude
  - Weight
  - Standard deviation

$$\omega_{k,t} = \alpha M_{k,t} + (1-\alpha)\omega_{k,t-1}$$

- α is the so-called learning rate
- M is **1** for the matching model and **0** otherwise

  □ if it is not the matching model, the weight is decreased

Exercise

- Go to OpenCV 4 documentation
- https://docs.opencv.org/4.x/
- Check the parameters for the `BackgroundSubtractorMOG`
- Try to change the number of Gaussians and the history (how much time you want to spend to learn the background model) used and check the results

---

- Change the MOG to `BackgroundSubtractorMOG2`
- Use the method `getBackgroundImage()` to get the background
- Display the background and observe how it changes over time with different values of the learning rate parameter