Computer Vision & Multimedia Analysis Course — A.A. 2019/2020

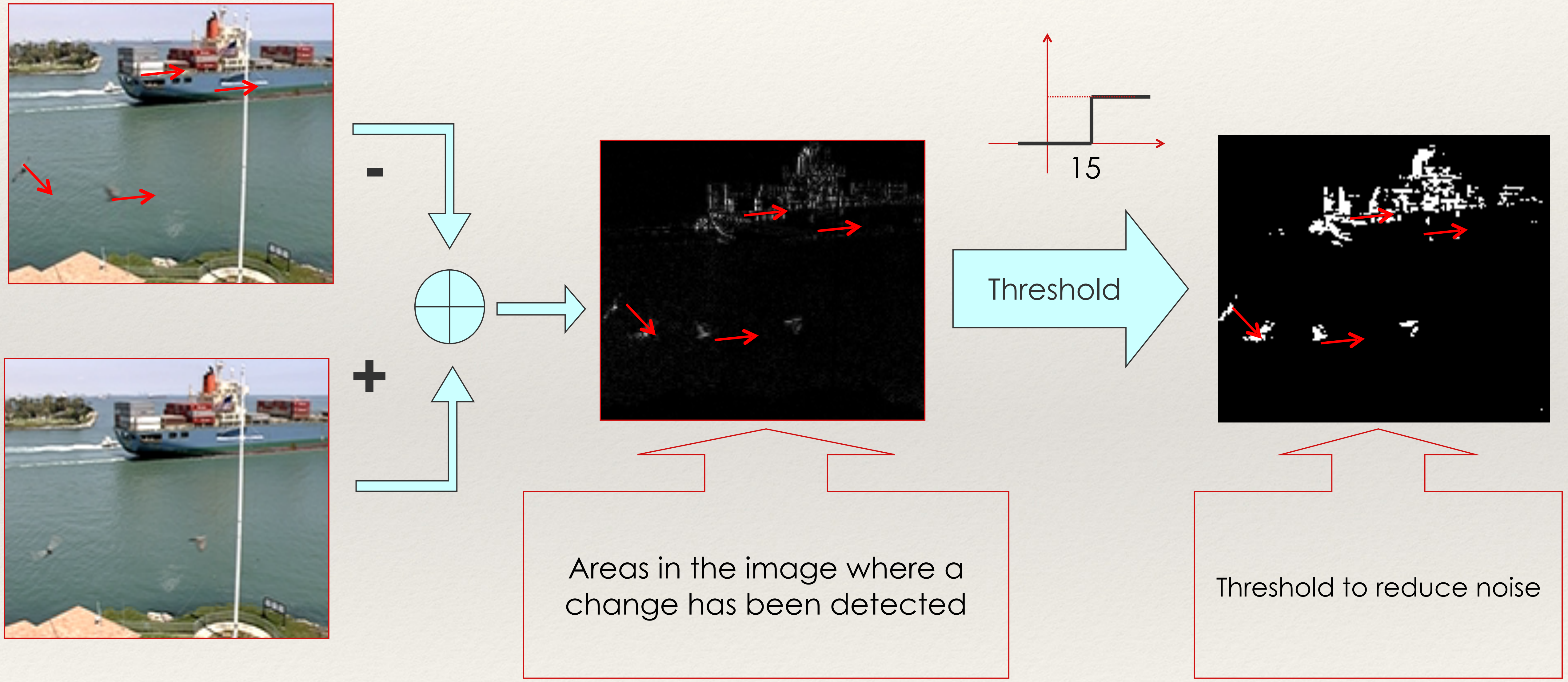
Lab 2: Motion Detection

Niccolò Bisagno
niccolo.bisagno@unitn.it

What's up today (and tomorrow)

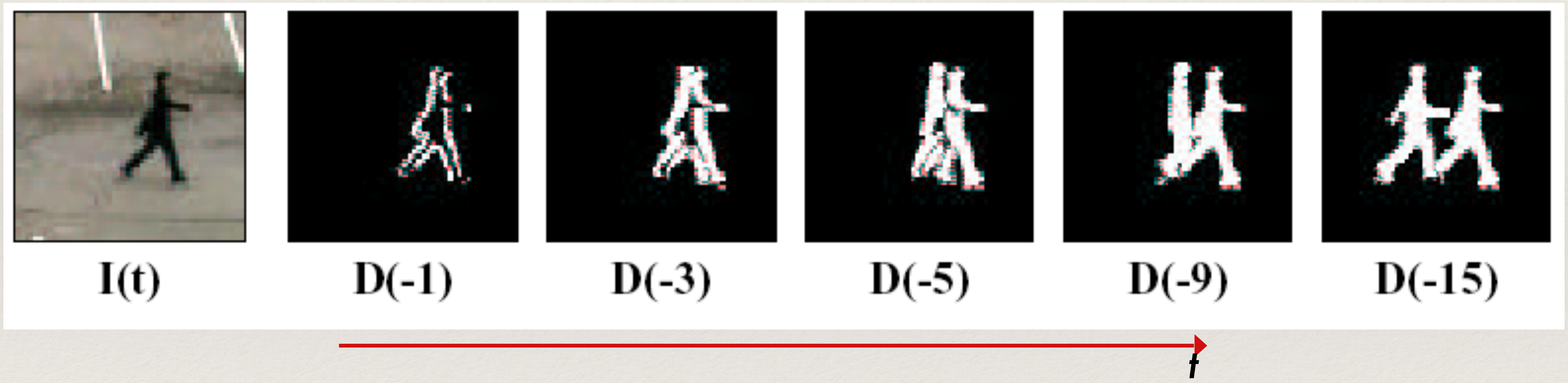
- Frame differencing
- Background Subtraction
- Adaptive Background Subtraction
- Adaptive Background Subtraction: Mixture of Gaussians

Frame differencing


-
+
15
Threshold
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

  cvtColor(frame_color,frame_gray, CV_RGB2GRAY);
- Use an array to store frames!

  Mat* frames = new Mat[1000] ;

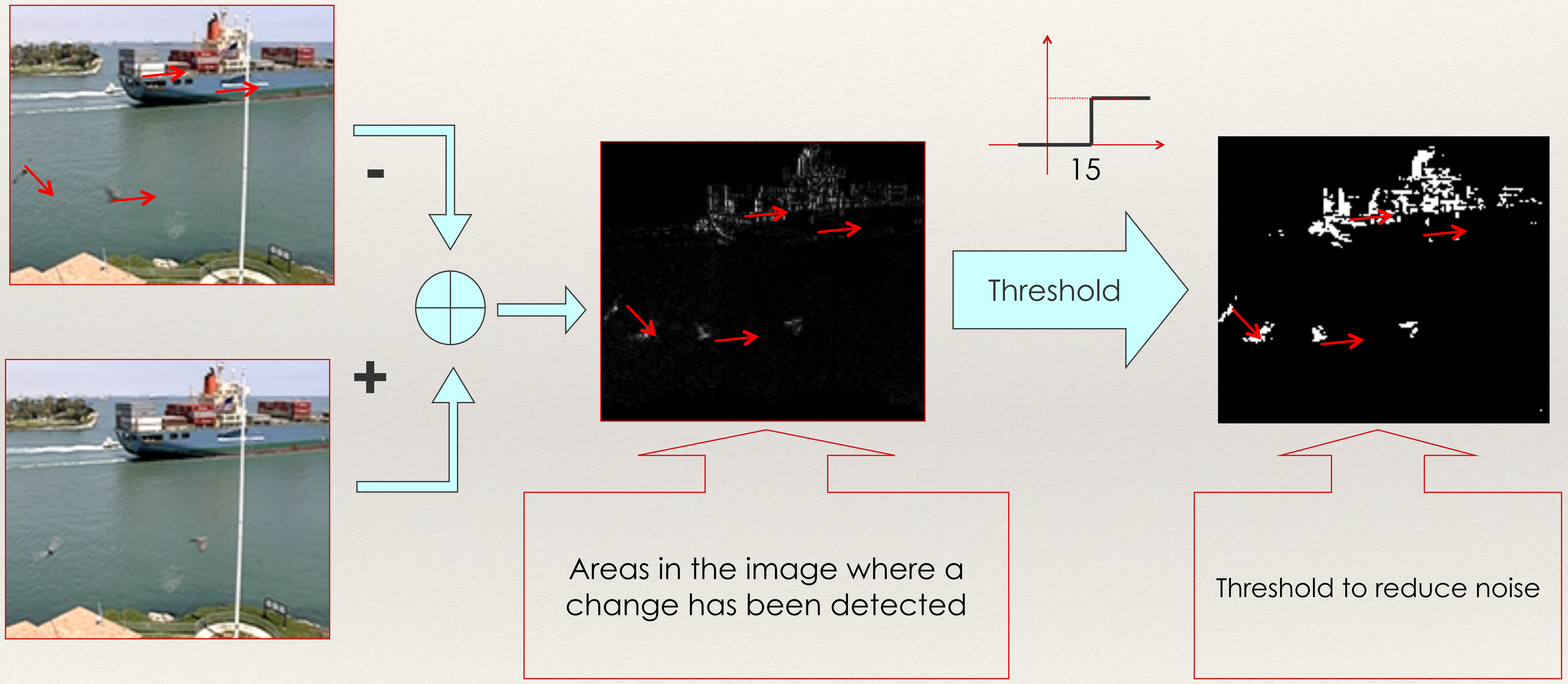
  NB!!!: how to copy: frame_gray.copyTo(frames);

  NB!!!: deallocate memory before return 0; -> delete[] frames;
- Implement the function

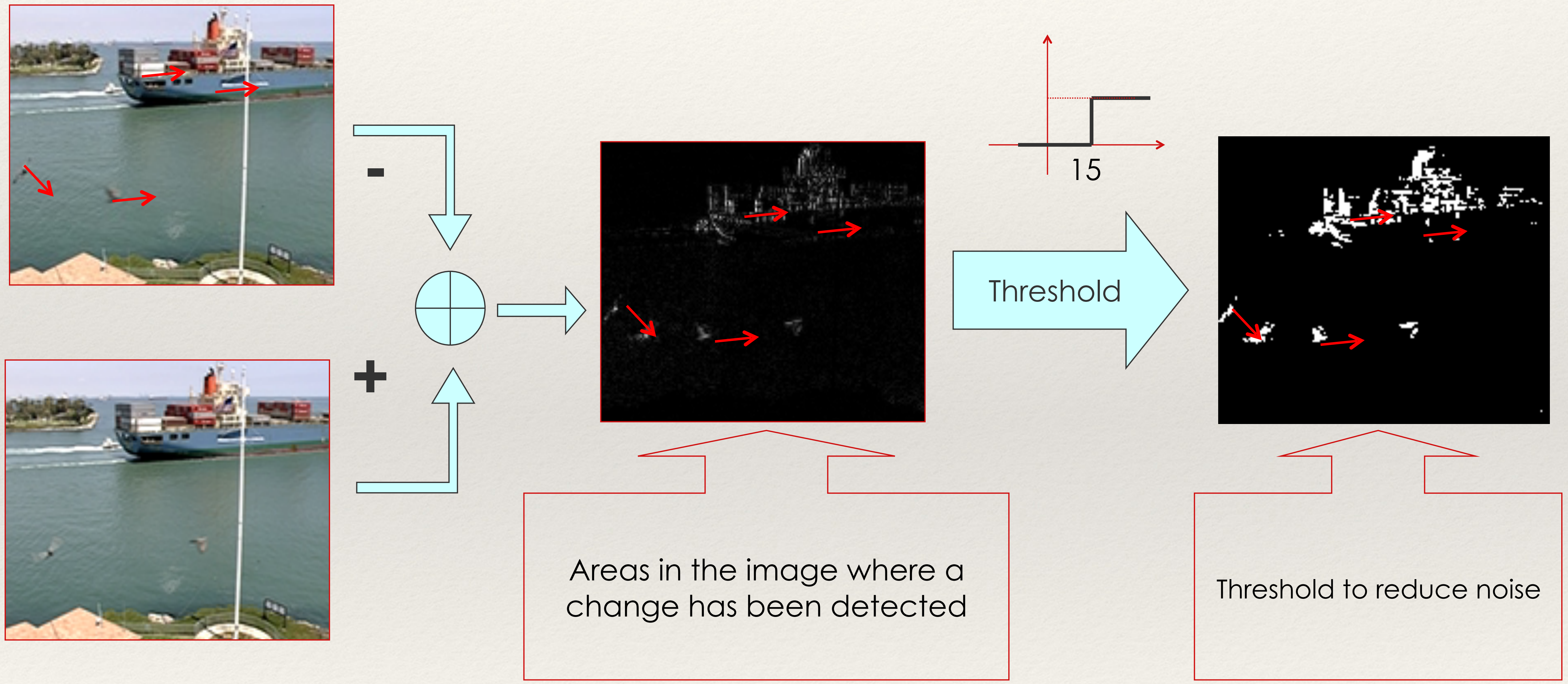
  absdiff(I(t),I(t+N),result);

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

What's missing in the implementation?


-
+
15
Threshold
Areas in the image where a change has been detected
Threshold to reduce noise

Apply thresholding on the mask


-
+
15
Threshold
Areas in the image where a change has been detected
Threshold to reduce noise

- threshold(InputArray src, OutputArray dst, double thresh, double maxval, int type)

Adaptive Background Subtraction

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

Mixture of Gaussians

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

- $\omega_{i,t}$ = 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 one 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/master/
- 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 (!!! MOG 2 doesn't need *bgsegm*)
- Use the method pMOG->getBackgroundImage(Mat bg); to get the background
- Display the background and observe how it changes over time with different values of the learning rate parameter