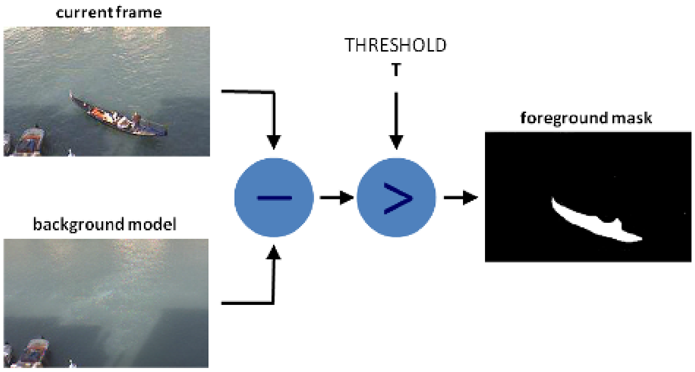
How to Use Background Subtraction Methods

Prev Tutorial: **High level stitching API (Stitcher class)**

Next Tutorial: **Meanshift and Camshift**

Original author	Domenico Daniele Bloisi
Compatibility	OpenCV >= 3.0

- Background subtraction (BS) is a common and widely used technique for generating a foreground mask (namely, a binary image containing the pixels belonging to moving objects in the scene) by using static cameras.
- As the name suggests, BS calculates the foreground mask performing a subtraction between the current frame and a background model, containing the static part of the scene or, more in general, everything that can be considered as background given the characteristics of the observed scene.



- Background modeling consists of two main steps:

  1. Background Initialization;
  2. Background Update.

  In the first step, an initial model of the background is computed, while in the second step that model is updated in order to adapt to possible changes in the scene.

- In this tutorial we will learn how to perform BS by using OpenCV.

Goals

In this tutorial you will learn how to:

1. Read data from videos or image sequences by using **cv::VideoCapture** ;
2. Create and update the background model by using **cv::BackgroundSubtractor** class;
3. Get and show the foreground mask by using **cv::imshow** ;

Code

C++ Java Python

In the following you can find the source code. We will let the user choose to process either a video file or a sequence of images.

We will use **cv::BackgroundSubtractorMOG2** in this sample, to generate the foreground mask.

The results as well as the input data are shown on the screen.

- **Downloadable code**: Click here
- **Code at glance:**

```
from __future__ import print_function
import cv2 as cv
import argparse

parser = argparse.ArgumentParser(description='This program shows how to use background subtraction methods provided by \
                                              OpenCV. You can process both videos and images.')
parser.add_argument('--input', type=str, help='Path to a video or a sequence of image.', default='vtest.avi')
parser.add_argument('--algo', type=str, help='Background subtraction method (KNN, MOG2).', default='MOG2')
args = parser.parse_args()
```

```
if args.algo == 'MOG2':
    backSub = cv.createBackgroundSubtractorMOG2()
else:
    backSub = cv.createBackgroundSubtractorKNN()



capture = cv.VideoCapture(cv.samples.findFileOrKeep(args.input))
if not capture.isOpened():
    print('Unable to open: ' + args.input)
    exit(0)


while True:
    ret, frame = capture.read()
    if frame is None:
        break


    fgMask = backSub.apply(frame)



    cv.rectangle(frame, (10, 2), (100,20), (255,255,255), -1)
    cv.putText(frame, str(capture.get(cv.CAP_PROP_POS_FRAMES)), (15, 15),
               cv.FONT_HERSHEY_SIMPLEX, 0.5 , (0,0,0))



    cv.imshow('Frame', frame)
    cv.imshow('FG Mask', fgMask)


    keyboard = cv.waitKey(30)
    if keyboard == 'q' or keyboard == 27:
        break
```

Explanation

C++ Java Python

We discuss the main parts of the code above:

- A **cv::BackgroundSubtractor** object will be used to generate the foreground mask. In this example, default parameters are used, but it is also possible to declare specific parameters in the create function.

```
#create Background Subtractor objects
if args.algo == 'MOG2':
    backSub = cv.createBackgroundSubtractorMOG2()
else:
    backSub = cv.createBackgroundSubtractorKNN()
```

- A **cv::VideoCapture** object is used to read the input video or input images sequence.

```
capture = cv.VideoCapture(cv.samples.findFileOrKeep(args.input))
if not capture.isOpened():
    print('Unable to open: ' + args.input)
    exit(0)
```

- Every frame is used both for calculating the foreground mask and for updating the background. If you want to change the learning rate used for updating the background model, it is possible to set a specific learning rate by passing a parameter to the `apply` method.

```
    #update the background model
    fgMask = backSub.apply(frame)
```

- The current frame number can be extracted from the **cv::VideoCapture** object and stamped in the top left corner of the current frame. A white rectangle is used to highlight the black colored frame number.

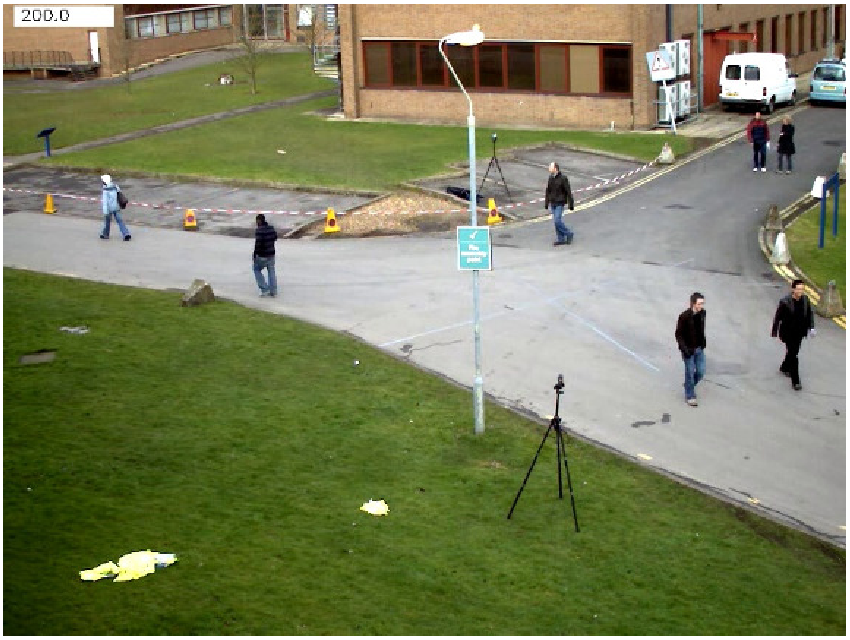
```
    #get the frame number and write it on the current frame
    cv.rectangle(frame, (10, 2), (100,20), (255,255,255), -1)
    cv.putText(frame, str(capture.get(cv.CAP_PROP_POS_FRAMES)), (15, 15),
               cv.FONT_HERSHEY_SIMPLEX, 0.5 , (0,0,0))
```

- We are ready to show the current input frame and the results.

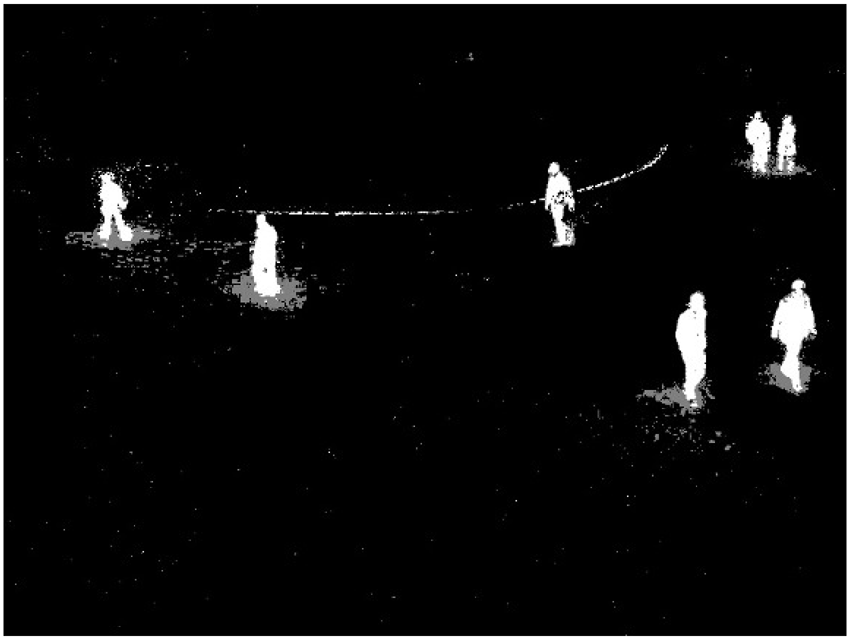
```
    #show the current frame and the fg masks
    cv.imshow('Frame', frame)
    cv.imshow('FG Mask', fgMask)
```

Results

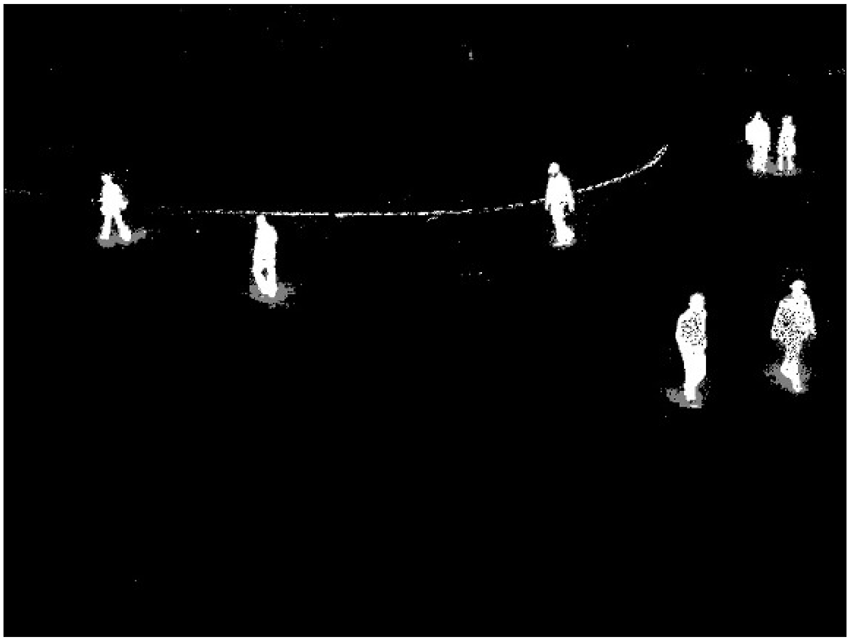
- With the `vtest.avi` video, for the following frame:



The output of the program will look as the following for MOG2 method (gray areas are detected shadows):



The output of the program will look as the following for the KNN method (gray areas are detected shadows):



References

- Background Models Challenge (BMC) website
- A Benchmark Dataset for Foreground/Background Extraction **[258]**

Generated on Sat Mar 26 2022 02:57:19 for OpenCV by doxygen 1.8.13