

Joint Master Degree (JMD) in
Image Processing and Computer Vision (IPCV)



Applied Video Sequence Analysis

Lab 1 “Foreground segmentation”

José M. Martínez - Paula Moral

josem.martinez@uam.es - paula.moral@uam.es

- **To develop a basic foreground segmentation algorithm** to detect moving objects on a stationary background
- **To improve the performance** of the basic algorithm
 - With additional techniques: running average background model update, suppression of stationary objects that appear/disappear, shadow removal
- **To develop “advanced” foreground segmentation algorithms using parametric models**
- **To evaluate the performance of foreground segmentation** algorithms using standard measures (Matlab code provided)
- **To write a report** to summarize achievements and results
 - Motivate the development of the different versions and modes (i.e., what does it enhance?)

- Assignment available on Moodle to submit your material
- The material must be submitted as a ZIP file with the following format *Name1Surname1_Lab1Task1.zip* or *Name1Surname1Name2Surname2_Lab1.zip*
- The submitted ZIP file will contain
 - Report in PDF format (max 6 pages)
 - *Makefile* to compile and link the program by simply running *make** (*Suggestion: use the makefile provided for Lab0*)
 - “src” directory with all source files (.h, .hpp, .c and .cpp) necessary for compiling and executing the corresponding program in Linux
 - “res” directory with the foreground masks for the *Baseline* category of CDNet2012 for different algorithms versions implemented.

*You may want to develop your C/C++ program using Eclipse but only the Makefile must be submitted (please do not submit Eclipse’s config files)

- Task 0 – Void foreground segmentation using template
 1. Go to “LAB 1” on Moodle
 2. Download the example project “LAB 1 code template.zip”
 3. Download the datasets “AVSASlidesVideosdataset” and “ChangeDetection dataset 2012 (baseline category only)”
 4. Uncompress the files
 5. Create a new Project Lab1.0AVSA2020 in Eclipse (see next slide)
 6. Open the Lab1.0AVSA2020.cpp file in the “src” directory
 7. Set the directory of the dataset and results according to your installation
 8. Compile it (“Build Project” in Eclipse)
 9. Create a “Launch Configuration” for the Project
 - Select the Project in the “Project Explorer” tab, go to the toolbar and select in the “Launch Configuration”pane “Create New Launch Configuration”
 - In the “Environment” tab set
LD_LIBRARY_PATH=/opt/installation/OpenCV-3.4.4/lib
 10. Run it ...

- Creating an Eclipse Project from scratch (see Lab0)
 - How to configure Eclipse CDT for using OpenCV3.4.4
- Creating an Eclipse Project with existing code
 - After creating the Projects and src folder ...
 - ... instead of creating a new source file (step 7), copy the files (.cpp, .hpp) into the src folder
 - Select them in a Files explorer and move them to the Project src folder in Eclipse
 - Continue with step 8
- ...

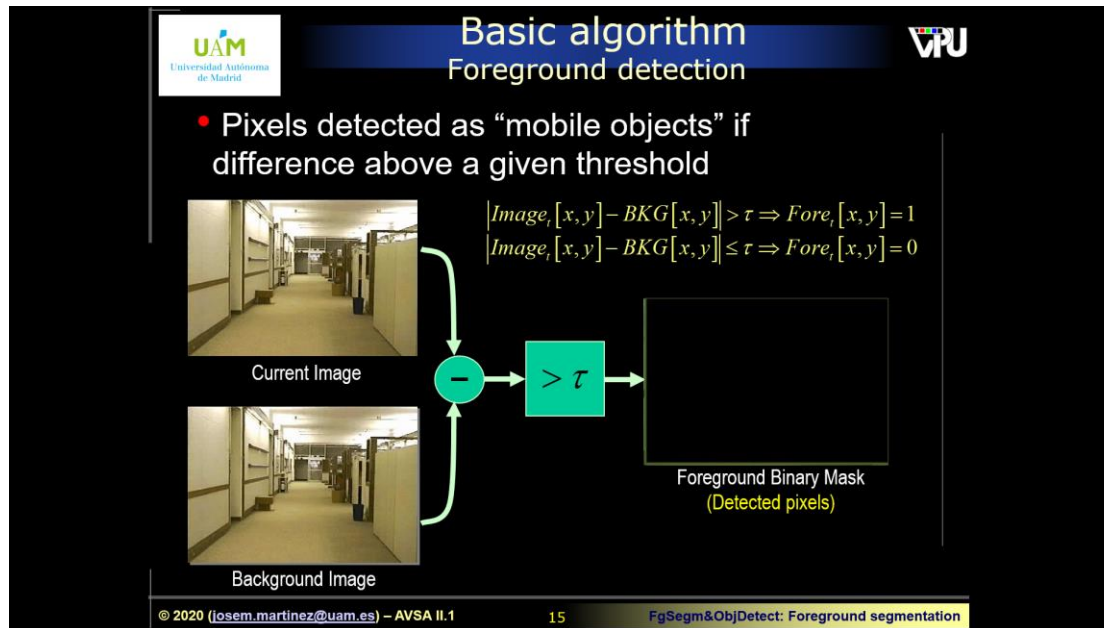
- ...
- Creating a copy of an Eclipse Project (for incremental development – e.g., different versions in Lab1)
 - Select source Project in “Project Explorer”: then “copy” and “paste”
 - Name the new project
 - Rename “main” file and change *project dependent* code (e.g., `project_name`)
 - Delete Binaries in Project Folders (e.g., Debug)
 - Build the Project
 - Create Launch Configuration for the new project (see Lab0)
 - Remember to set `LD_LIBRARY_PATH`
 - Run the Project

- Modifying the Lab0 Makefile (for code submission) *to include (using the code of this task as example) the fgseg.cpp and ShowManyImages.cpp files*
 - *So, when we include cpp files in our project, we have to add in the Makefile the target(s) to generate the object(s) :*
`fgseg.o: fgseg.cpp`
`g++ -c fgseg.cpp -I$(PATH_INCLUDES) -O`
 - *and then to execute all the objects created and generate the executable, we have to indicate all the dependencies:*
`all: Lab1.0AVSA2020`
`Lab1.0AVSA2020: Lab1.0AVSA2020.o fgseg.o ShowManyImages.o`
`g++ Lab1.0AVSA2020.o fgseg.o ShowManyImages.o -o`
`Lab1.0AVSA2020 -L$(PATH_LIB) $(LIBS) -lm`

- Task 1 - Implementation of foreground segmentation algorithms using the OpenCV library

–Methods


- **Lab1.1.1: Generation of foreground segmentation mask for each video frame using “frame difference”**
 - Parameters: tau (double), rgb (true/false)
 - *Recommendation: firstly, implement gray mode*




- Task 1 - Implementation of foreground segmentation algorithms using the OpenCV library

–Methods

- **Lab1.1.2: Progressive update of background model through selective running average (blind and selective update)**
 - Parameters: tau (double), rgb (true/false), alpha (double), selective_update (true/false)
 - *Recommendation: firstly, implement blind mode*



Background model update



- Progressive adaptation of background model:
 - Background image updated with part of current image for pixels not belonging to the foreground mask ➡ **selective running average**

$$BKG_{t+1}[x,y] = \alpha Image_t[x,y] + (1-\alpha)BKG_t[x,y] \quad \longleftrightarrow \quad Fore_t[x,y] = 0$$

- Parameter α determines adaptation speed (e.g., $\alpha = 0.05$)

© 2020 (josem.martinez@uam.es) – AVSA II.1

34

FgSegm&ObjDetect: Foreground segmentation

- Task 1 - Implementation of foreground segmentation algorithms using the OpenCV library

—Methods

- **Lab1.1.3: Suppression of stationary objects that appear or are removed**
 - Parameters: tau (double), rgb (true/false), alpha (double), threshold (int)

The screenshot shows a presentation slide with a dark background. At the top left is the UAM logo. At the top center is the title 'Background model update' in a blue box. At the top right is the VPU logo. The main content is a bulleted list describing the suppression of stationary objects. At the bottom, there is a footer with copyright information, a slide number, and a title for the current slide.

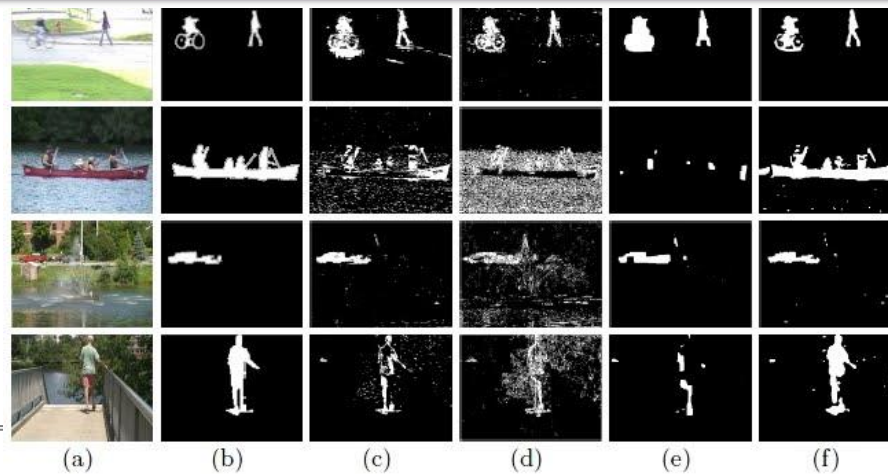
Background model update

- Suppression of ghosts (2) - stationary objects that appear or are removed:
 - Counter per pixel
 - Counter incremented every time pixel is detected as foreground in current image
 - Counter reset every time pixel is detected as background in current image
 - Pixel “incorporated” to background model if counter reaches a threshold

© 2020 [josem.martinez@uam.es] – AVSA II.1 37 FgSegm&ObjDetect: Foreground segmentation

- Task 1 - Implementation of foreground segmentation algorithms using the OpenCV library
 - Requirements
 - Only use three source files (main.cpp, fgseg.cpp and fgseg.hpp) for each project (Lab1.1.1, Lab1.1.2 and Lab1.1.3)
 - The algorithm operates over gray-level images and color-images
 - Report comparative subjective results using the VideoSlidesAVSA dataset for the three versions
 - Report a sensitivity analysis of the parameters involved in algorithms (i.e., study the values of the parameters involved and their impact in the output or the foreground mask)
 - Recommendations
 - Develop firstly the gray-level versions
 - Motivate in the Report the development of the different versions and modes (i.e., what does it enhance?)
 - Provide a subjective comparative among versions using the AVSASlidesVideo dataset


- Task 2 – Evaluate your (***selected***) algorithm using the ChangeDetection datasets (2012 version, CDNet2012)
 - Evaluate the algorithm developed in Task 1 using the Baseline category from <http://changedetection.net/>
 - Category of the dataset available in Moodle
 - MATLAB code is available for evaluating the results of your algorithms
 - In the report, include performance comparisons with selected approaches from <http://jacarini.dinf.usherbrooke.ca/results2012>



• Task 3.1 - Implementation of a shadow suppression module using the OpenCV library


—Method

- Suppression of shadows based on the chromaticity-based method


Universidad Autónoma de Madrid

Basic algorithm

Chromaticity-based method



- Shadow detection

$$Image_t[x, y] = (IH_t[x, y], IS_t[x, y], IV_t[x, y])$$

$$BKG_t[x, y] = (BH_t[x, y], BS_t[x, y], BV_t[x, y])$$

$$\alpha \leq \frac{IV_t[x, y]}{BV_t[x, y]} \leq \beta \wedge |IS_t[x, y] - BS_t[x, y]| \leq \tau_S \wedge D_H \leq \tau_H \Rightarrow Shadow_t[x, y] = 1$$

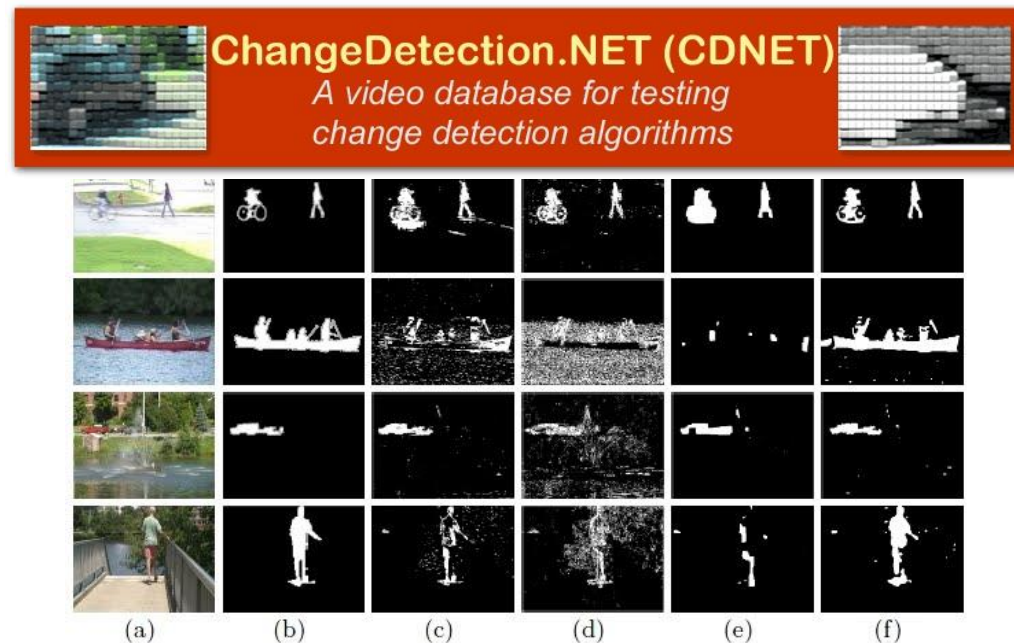
$$D_H = \min(|IH_t[x, y] - BH_t[x, y]|, 360 - |IH_t[x, y] - BH_t[x, y]|)$$

(e.g., $\alpha = 0.5, \beta = 0.9$)

- Thresholds determined empirically

© 2019 (josem.martinez@uam.es) – AVSA II.2
108
FgSegm&ObjDetect: Shadow detection

- Task 3.2 – Evaluate the suppression of shadows method
 - Evaluation of new algorithm using the *Shadow* category from <http://changedetection.net/>




• Task 4.1 – Implementation of advanced background subtraction algorithms

–Method

- Single Gaussian


–Recommendation

- Implement (mandatory) the gray-level version (but shadows suppression ...)



Robust background models

Parametric models



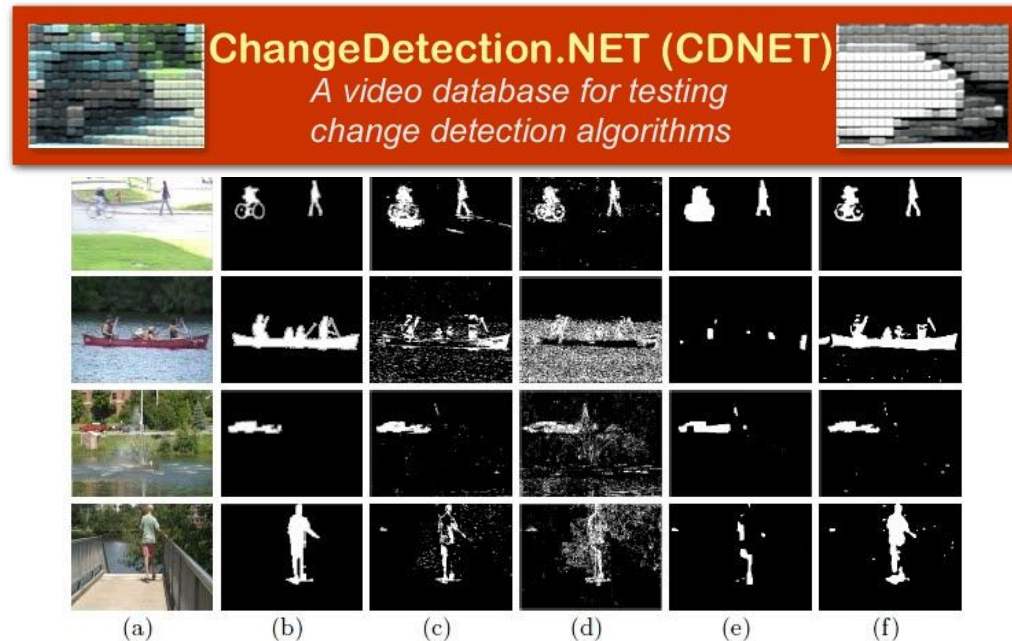
- Unimodal model (Gaussians)
 - Background model (Gaussian distribution for each pixel) → Mean and standard deviation per pixel $\mu_i[x, y], \sigma_i[x, y]$

© 2020 (josem.martinez@uam.es) – AVSA II.1

49

FgSegm&ObjDetect: Foreground segmentation

- Task 4.2 – Evaluation of advanced background subtraction algorithm
 - Evaluation of the new algorithm using the *Dynamic Background* category from <http://changedetection.net/>




• Task 5.1 – Implementation of advanced background subtraction algorithms

–Method

- GMM

–Recommendation


- Implement (mandatory) the gray-level version (but shadows suppression ...)



UAM
Universidad Autónoma
de Madrid

Robust background models

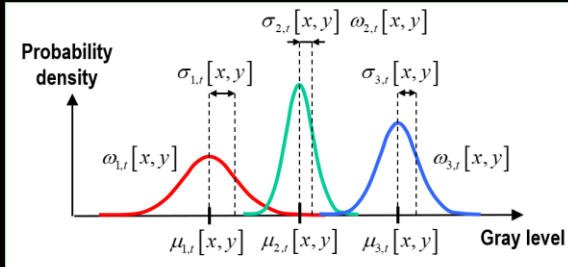
Parametric models



VPU

- Multimodal model (Mixture of Gaussians)
 - Background model (*K* Gaussian distributions per pixel; *K* between 3 and 5) ➔ Mean, deviation and weight per Gaussian

$$\mu_{i,t}[x, y], \sigma_{i,t}[x, y], \omega_{i,t}[x, y]$$

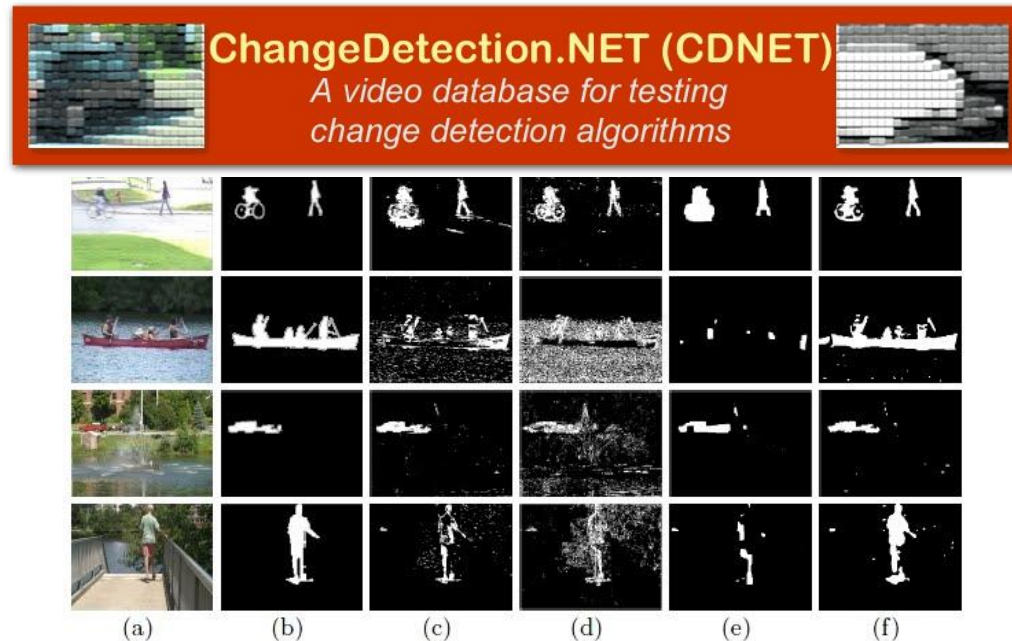


© 2020 (josem.martinez@uam.es) – AVSA II.1

52

FgSegm&ObjDetect: Foreground segmentation

- Task 5.2 – Evaluation of advanced background subtraction algorithm
 - Evaluation of the new algorithm using the *Dynamic Background* category from <http://changedetection.net/>

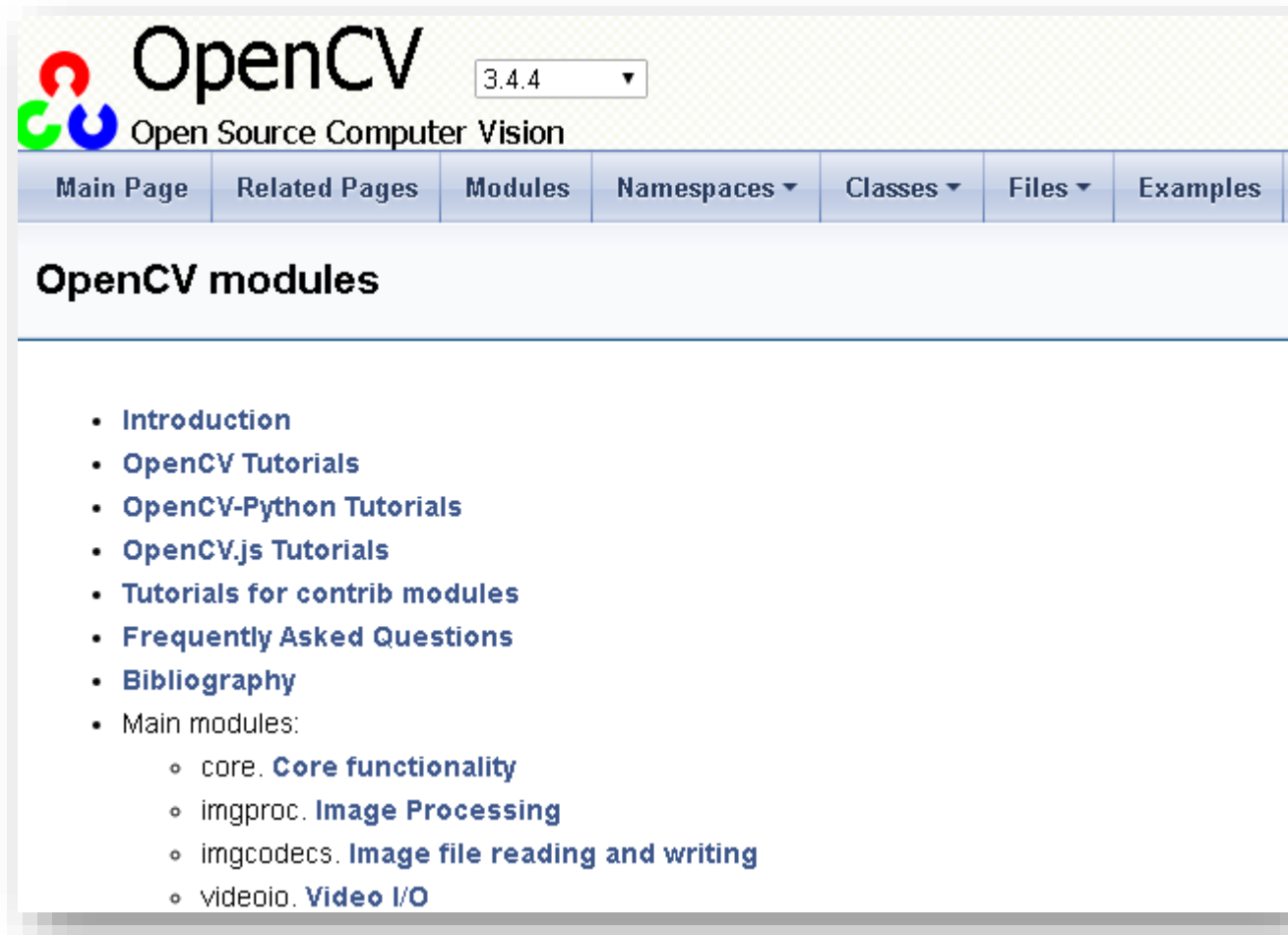


- This lab assignment will be **graded with 10 points**
 - Task 1 (up to 3 points)
 - Individual work
 - Submission should include a short report (free format) including comparative subjective results and a sensitivity analysis using the VideoSlidesAVSA dataset for the three versions
 - Task 2 (up to 1 points)
 - Task 3 (up to 1 points)
 - Task 4 (up to 1 points)
 - Task 5 (up to 2 points)
 - [an extra point may be awarded for a working color version in Tasks 4 and 5]
 - Report (up to 2 points) – Please follow the report format (see *Moodle*)
 - The report should include a section describing the selection (rationale) of one algorithm between the two (individual Task 1) of the group members

- The following general criteria will be used for grading:
 - The solution addresses the requirements for the assignment.
 - The program compiles, links, and executes.
 - The program runs correctly.
 - The program is easy to read and to understand, i.e., it is well commented, and variables are correctly named.
 - The lab report clearly describes the algorithm and experiments.
- The following specific criteria will be used for grading:
 - Foreground segmentation using colour information.
 - Complexity of the implemented algorithms
 - Processing with OpenCV functions at matrix/vector level instead of pixel-level.

In addition, up to an extra point (1 point) may be awarded depending on the quality of the software delivered in the assignment.

- Check OpenCV documentation for finding specific functions at <https://docs.opencv.org/3.4.4/>



The screenshot shows the OpenCV 3.4.4 documentation website. At the top, there is the OpenCV logo (three interlocking circles in red, green, and blue) followed by the text "OpenCV" and "Open Source Computer Vision". To the right of the logo is a dropdown menu showing "3.4.4". Below this is a navigation bar with tabs: "Main Page", "Related Pages", "Modules", "Namespaces", "Classes", "Files", and "Examples". The "Modules" tab is selected. Below the navigation bar, the heading "OpenCV modules" is displayed. Under this heading, there is a list of links: "Introduction", "OpenCV Tutorials", "OpenCV-Python Tutorials", "OpenCV.js Tutorials", "Tutorials for contrib modules", "Frequently Asked Questions", "Bibliography", and "Main modules:". Under "Main modules:", there are four sub-items: "core. Core functionality", "imgproc. Image Processing", "imgcodecs. Image file reading and writing", and "videoio. Video I/O".