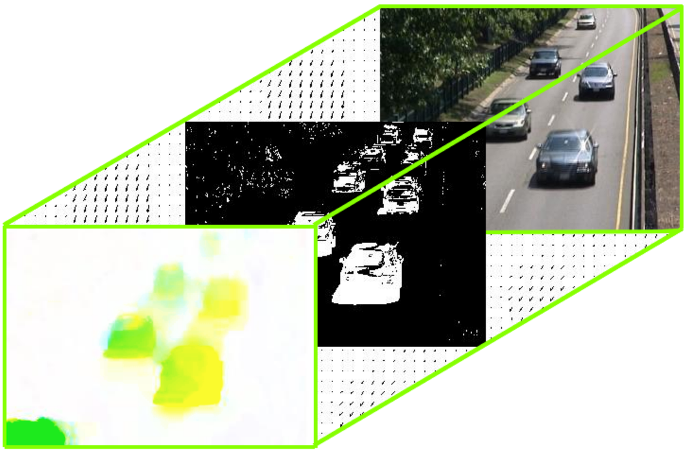
MASTER IN COMPUTER VISION BARCELONA

VIDEO SURVEILLANCE FOR ROAD TRAFFIC MONITORING

MODULE 4: VIDEO SEQUENCE ANALYSIS



Authors

JOSEP RAMON CASAS	—	josep.ramon.casas@upc.edu
LESLEY-ANN DUFLOT	—	lesleyann.duflot@gmail.com
XAVIER GIRÓ	—	xavier.giro@upc.edu
MONTSE PARDÀS	—	montse.pardas@upc.edu
JAVIER RUIZ HIDALGO	—	j.ruiz@upc.edu
VERÓNICA VILAPLANA	—	veronica.vilaplana@upc.edu

COORDINATOR: JAVIER RUIZ HIDALGO & XAVIER GIRÓ

01 NOVEMBER 2013

Abstract

The goal of this project is to learn the basic concepts and techniques related to video sequences processing, mainly for surveillance applications. We will focus on video sequences from outdoor scenarios, with the application of traffic monitoring in mind. The main techniques of video processing will be applied in the context of video surveillance: moving object segmentation, motion estimation and compensation and video object tracking are basic components of many video processing systems. In a first stage, moving object segmentation will be tackled considering scenarios with static camera. Afterwards, camera motion will be considered. Tracking of the moving objects can be performed in both scenarios. The tracking result provides high level information that can be analysed for traffic monitoring. The learning objectives for the students are the use of pixel based statistical models (such as mixture of gaussians) for modeling a scene background and for moving object segmentation, the development of optical flow estimation methods for camera motion compensation, and techniques for object tracking (ranging from simple blob analysis to more complex techniques based on filtering and probabilistic data association). The performance of the developed techniques will be measured using standard metrics for video analysis.

Table of Contents

1

Project Overview

The goal of this project is to learn the basic concepts and techniques related to video sequences processing, mainly for surveillance applications. We will focus on video sequences from outdoor scenarios, with the application of traffic monitoring in mind. A working prototype to segment cars from surveillance videos will be implemented through the project. The prototype will be robust againts:

- illumination changes in the video,
- noise and degraded images (such as the resulting from low bitrate coding),
- unwanted camera vibrations.

1.1 Project Stages

The whole task of the project is organized in the following stages, namely:

- *Background estimation*
- *Foreground segmentation*
- *Video stabilization*
- *Region tracking*

This will allow to deal with its complexity and to favor the understandability of the techniques employed.

Figure 1.1 represents a diagram of the main tasks of the project.

1.1.1 Background estimation

The goal of this submodule is to model the background information of a video sequence. In a static camera sequence, in order to segment the moving objects, a good estimate of the background is needed. The approach will be to create a pixel based statistical model of the background.

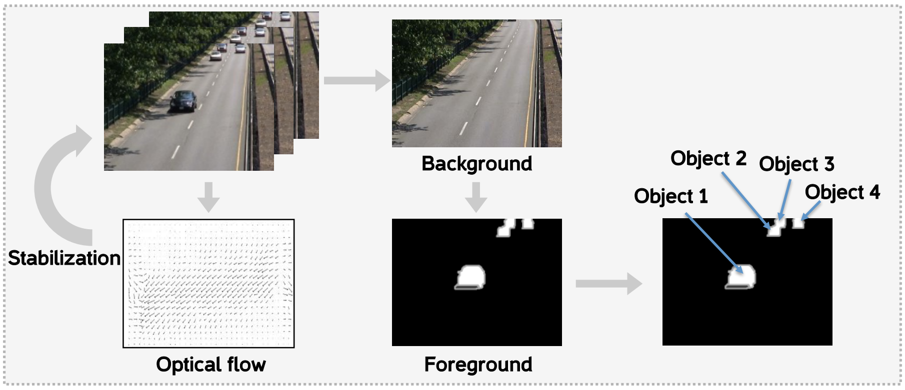


Figure 1.1: Project overview

1.1.2 Foreground segmentation

Once we have generated an estimation of the background in the previous submodule, the goal now is to segment the moving objects of the video. The approach to deal with this purpose is to compare the pixel values of new frames with the corresponding background model and treat as foreground all pixels that do not fit in the estimated model.

The above classification could be noisy, additional stages need to be included to improve the above solution are:

- A connected component analysis and filtering to remove spurious detection due to the pixel based approach.
- A shadow removal system as shadow are usually detected as foreground.

1.1.3 Video stabilization

Previous steps may fail in real conditions due to the movement of the camera. This movement could be caused by unwanted camera vibrations (due to the wind in outside placed cameras) or by controlled motion of the cameras (such as surveillance cameras that sweep a region under supervision). This stage will be approached with a motion estimation and compensation technique. The motion estimation will consists on an optical flow computation with state of the art techniques. The optical flow will be used to stabilize the unwanted movement of the video sequence creating a new version of the video sequence (compensated sequence). Finally, the performance of the previous stages will be evaluated with the compensated sequence.

1.1.4 Region tracking

The goal of this module is to track the objects extracted during the foreground segmentation stage to uniquely identify the same object through the entire sequence. This will allow extracting the trajectories of the moving objects. These trajectories will be used for a semantic analysis of the video sequence.

The approach used in the region tracking will include a Kalman filter to predict the trajectories of each object.

1.2 Learning goals

To solve this project you will learn about:

Statistical models Statistical models will be studied and applied to model the color pixels of the background.

Connected component analysis Area filters and hole filling techniques will be studied and applied to the problem of foreground segmentation.

Color models Different color models (RGB, YUV) will be applied in the shadow removal of foreground objects.

Optical flow Principles and limitations of optical flow estimation techniques. Algorithms for optical flow computation that can be used for global motion estimation: feature-based and featureless (local and global) techniques.

Object Tracking Ranging from simple blob analysis to more complex techniques based on filtering and probabilistic data association.

Evaluation metrics Commonly used evaluation metrics for foreground segmentation and optical flow estimation will be studied and implemented.

2 Assignments

The development of the whole system has been planned in 5 weeks, as presented in detail in this Section.

2.1 Week 1. Assessment of Foreground Extraction and Optical Flow

The goals of this week is to understand and familiarize with the programing framework used in this project. Furthermore, the datasets used in the project together with the evaluation metrics that measure the performance of the algorithms used in the project will be implemented and preliminary tested.

You will create some functions to open and write the video sequences provided by the datasets. You will familiarize with the ground-truth provided by the databases and create functions to evaluate the results by measuring the proposed evaluation metrics.

2.1.1 Mandatory tasks

Foreground Estimation Metrics

You are provided with two test sequences: *test A* and *test B* which correspond to a foreground estimation of the "highway" sequence in the Change Detection Video Database. This estimation has been produced by using a background model of 3 gaussians with different thresholds for the *A* and *B* test sequences respectively. The *test A* sequence aims at a high recall and the *test B* sequence at a high precision. Both sequences are composed of 200 frames and correspond to the foreground estimation of the sequence "highway" between frames 1201 and 1400. Please take notice that there are different levels of grey in the ground-truth and test sequences. In the test sequences, we use a gray value of 1 if the pixel is in the foreground and 0 if the pixel is in the background.

Task 1 With the corresponding ground-truth of the "highway" sequence (in the baseline category), calculate the number of: False Negative, True Negative, False Positive and True Positive, the Precision, the Recall and the F1 score for the *200 frames* of both *test A* and *test B* sequences. It would be very useful if you can make a function (or functions) to perform those calculations because it will be very useful for next week assignments. Present the obtained values in a Table like the following:

	test A	*test B*
True Positive		
True Negative		
False Positive		
False Negative		
Precision		
Recall		
F1 score (F-measure)		

Task 2 Represent the foreground estimation of a single frame, for instance, frame number 1300, for both test sequences and try to explain why one of the has higher recall / precision than the other one.

Task 3 Instead of computing the metrics for all the sequence (as in Task 1), compute them on a frame by frame basis and create a plot to compare the number of True Positive values and the foreground pixels in the ground-truth frame by frame. Do a similar plot (frame by frame) with the F1 score for both test sequences and comment any significant deviations of the F1 score for specific frames compared to the single value obtained for the entire sequence, as found in Task 1.

Motion Estimation Metrics

Two motion vector estimations are provided using the Lucas-Kanade optical flow algorithm for sequences 45 and 157 of the KITTI dataset. Only one of the stereo views is analyzed (*image_0*). We will use the ground truth without occluded estimations, referred as *flow_noc* in the dataset.

The files storing the estimations are:

- **LKflow_000045_10.png** estimates the optical flow between images *image_0/000045_10.png* and *image_0/000045_11.png* with ground truth *flow_noc/000045_10.png*.
- **LKflow_000157_10.png** estimates the optical flow between images *image_0/000157_10.png* and *image_0/000157_11.png* with ground truth *flow_noc/000157_10.png*.

For both estimations:

Task 4 Compute the mean magnitude error, MMEN, for all pixels in non-occluded areas.

Task 5 Calculate the percentage of erroneous pixels, PEPN, in non-occluded areas. Erroneous pixels are defined as pixels where the corresponding motion vector has an error greater than 3.

2.1.2 Optional tasks

Task 6 In video sequence analysis sometimes the obtained results are not synchronized with the ground truth. This can happen, for instance, when there is a delay in the computation. Evaluate the degradation due to problems in the synchronization step by plotting a curve of the F1 score computed in Task 1 but, this time, desynchronize the result and the ground truth. Make several measures displacing the results (1, 2, 3, 4, ... 25 frames) with respect to the ground truth and represent the curve in a single plot.

Task 7 To subjectively evaluate the optical flow, sometimes the motion vectors are plotted overlaid in the image as the following figure represents.

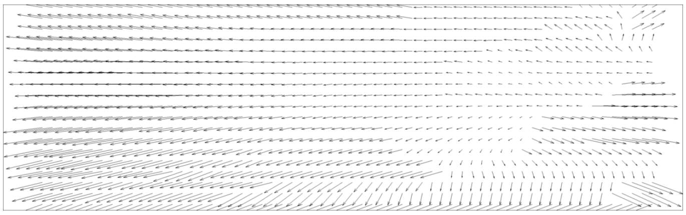


Figure 2.1: Motion vectors resulted from the optical flow estimation

Create some code in Matlab to represent the results given in Task 4 (use the function *quiver* in Matlab).

2.1.3 Performance evaluation

As mentioned in the tasks description, the evaluation will consists in two different types of output. Firstly, the evaluation of the foreground segmentation will be performed by comparing the ground truth and the provided results using the True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) values. Furthermore, the precision, recall and F1 score will be provided.

Secondly, the evaluation of the optical flow will be evaluated using the percentage of erroneous pixels in non-occluded areas (PEPN), the mean square error in non-occluded areas (MSEN).

2.1.4 Deliverable

Items to be delivered:

- Progress slides: A *short* power point presentation (∼5 min.) commenting the used approach, results obtained and conclusions.
- Source code: A working version of the Matlab code developed along the week. The code must be clearly commented and a README file must be included with instructions of how to use it.

2.2 Week 2. Background estimation

The goal of this week is to model the background pixels of a video sequence using a simple statistical model. Firstly, a single gaussian model per pixel will be used to model the background pixels. The statistical model will be used to preliminary classify foreground pixels in the sequence. The simple model will be compared with more complex ones such as the Stauffer and Grimson (provided in the programming framework).

For all the exercises in this week, we will work with a subset of 3 sequences from the Change Detection dataset converted to ***gray-scale***:

- frames 1050 to 1350 for *highway* sequence in the baseline category.
- frames 1460 to 1560 for *fall* sequence in the dynamic background category.
- frames 950 to 1050 for *traffic* sequence in the camera jitter category.

2.2.1 Mandatory tasks

Non-recursive Gaussian modeling

Task 1 Create a function that computes a one gaussian distribution to model the background pixels. Use the first 50% of the sequence to estimate the background mean μ_i and standard deviation σ_i for each pixel and the last 50% of the frames to classify foreground and background pixels by a simple threshold α:

for all pixels i **do**
  if $|I_i - \mu_i| \geq \alpha \cdot (\sigma_i + 2)$ **then** ▷ +2 to prevent low values of σ_i
    pixel → Foreground
  else
    pixel → Background
  end if
end for

where I_i represent the gray-scale value of the pixel i.

Task 2 Draw the curves F1 score, True Positive, True Negative, False Positive, False Negative vs. threshold α for the three proposed sequences (remember to convert them to gray-scale).

Task 3 Draw the curve Precision vs. Recall depending of threshold α for the three proposed sequences and comment the results.

Recursive Gaussian modeling

In this task, we will try to update the background model recursively. For that we will need a refresh rate ρ. Re-use the function in Task 1 and create a new recursive function where, at each step of the classification (in the latest 50% of the frames), update the background mean and standard deviation for the pixels that are classified as Background:

if pixel $i \in$ Background **then**
  $\mu_i = \rho \cdot I_i + (1 - \rho) \cdot \mu_i$

$\sigma_i^2 = \rho \cdot (I_i - \mu_i)^2 + (1 - \rho) \cdot \sigma_i^2$
end if

Task 4 Implement the recursive function described above and discuss which is the best value of ρ for the *fall* sequence.

Task 5 Compute the F1 score (for the fixed value of ρ computed in Task 4 and α in Task 3) and compare with the non-recursive version for the three proposed sequences and comment the results.

Stauffer and Grimson

Finally, we will compare the single gaussian modeling studied in the previous tasks with the Stauffer and Grimson multiple gaussian approach (S&G). You can find an implementation of the S&G method in the Computer Vision Toolbox in Matlab as the system object *vision.ForegroundDetector*. You can also find an alternative implementation in the provided Matlab code if you have not access to the Computer Vision Toolbox.

Task 6 Use the S&G approach and compute the F1 score for the three provided sequences using a different number of gaussians (from 3 to 6). Find out the optimal number and comment the results obtained.

Task 7 Compare your gaussian modeling of the Background pixels with S&G using the F1 score and comment which sequences benefit more from using more gaussians in the modeling.

2.2.2 Optional tasks

Task 8 Update the single gaussian functions (recursive and non-recursive) to work with color images and use them to obtain the F1 score of the three proposed sequences.

2.2.3 Performance evaluation

The evaluation will be based on the precision, recall and F1 score provided for the different sequences.

2.2.4 Deliverable

Items to be delivered:

- Progress slides: A *short* power point presentation (∼5 min.) commenting the used approach, results obtained and conclusions.
- Source code: A working version of the Matlab code developed along the week.

2.3 Week 3. Foreground segmentation

The goal for this week will be to implement a robust foreground segmentation algorithm. The statistical model implemented in the previous week is the baseline which must be improved during this week. Two basic strategies are to applied: hole filling and area filtering. In addition, a strategy to remove the shadows of the cars can also improve the final performance.

2.3.1 Mandatory tasks

This task continues the results obtained during the previous week, which are to be improved by applying new techniques. Thus, consider as an input for this task the foreground masks obtained with the best configurations of the previous week.

The foreground masks generated using only the modelling of the pixel values as Gaussian distributions produced a results which did not take into account the neighbourhood of each pixel. A first strategy to improve the performance is the application of tool of morphological filtering to obtain a cleaner signal based on simple operations. The first steps for a higher performance include hole filling and area filtering.

Task 1 Use the Matlab function *imfill* to fill the black holes inside white regions. Run your experiments by considering both a 4- and a 8-connectivity. Show the results on a table comparing the obtained Precision, Recall and F1-score with the baseline values obtained last week.

Task 2 Use the Matlab function *bwareaopen* to filter out those regions which are too small to be considered a car. Plot a graph relating the allowed minimum size with the obtained F1 score. The graph must plot four cases: (a) baseline results with no morphological filtering, (b) area filtering only, (c) area filtering + hole filling with connectivity 4 and (d) area filtering + hole filling with connectivity 8. Analyse the results of the graph.

Task 3 Generate one plot containig the Precision-Recall curves configurations (b), (c) and (d) from Task 2, and a single PR dot for case (a) . Analyse the results obtained.

2.3.2 Optional tasks

Task 4 The hole filling and area filtering are just two of the multiple operations available in morphological filtering. Explore other solutions that may improve the results already achieved with the hole filling and area filtering

Task 5 Develop an algorithm capable of removing the shadow of the car. Assess if your solution does improve previous results.

Task 6 Generation a foreground map, which is normalized version of the image whose pixels estimate the probability of belonging to the foreground class. Provide a numerical assessment of the quality of the foreground map by using, for example, the solution proposed in R. Margolin, L. Zelnik-Manor and A. Tal, “How to Evaluate Foreground Maps?” (CVPR 2014).

2.3.3 Performance evaluation

The performance metrics are the same as in week 2: maximum F1-score and Area Under the Curve of the Precision-Recall curve of the labelling of pixels between foreground and background.

2.3.4 Deliverable

Items to be delivered:

- Progress slides: A *short* power point presentation (∼5 min.) commenting the used approach, results obtained and conclusions.
- Source code: A working version of the Matlab code developed along the week. The code must be clearly commented and a README file must be included with instructions of how to use it.

2.4 Week 4. Video Stabilization

Surveillance cameras are often placed on physical support that can vibrate because of weather conditions or nearby transportations. This camera jitering affects the video sequence, introducing an motion on both foreground objects and background. As a result, analysis algorithms that depend from the extraction and tracking of foreground object can be contaminated by the camera motion. This module focuses in the estimation of such camera motion and existing techniques to achieve a virtual stabilization of the camera.

2.4.1 Mandatory tasks

Task 1 Implement a block matching solution for the estimation of the optical flow. You must decide what kind of compensation to use (backrward or forward), the area of search and the size of the blocks.

Task 2 Compare the generated optical flows with the results obtained with the Lucas-Kanade algorithm.

Task 3 Use the estimated flow between two frames to align them and apply it to stabilize the camera on the TRAFFIC scene. Assess again the foreground extraction algorithm after camera stabilization using your best configuration obtained in the previous week.

2.4.2 Optional tasks

Task 4 Search online for at least two other implementations of optical flow estimations and assess them on the provided sequences. Provide the source code and references in your submission.

Task 5 Search online for at least two other implementations of video stabilizations and assess them on the provided sequences. Provide the source code and references in your submission.

Task 6 Another classic case of undesired camera motion is the one introduced by non-professional users handling their personal cameras. This optional task requires recording a short video with a user's camera (eg. smartphone) and stabilize it whether with your own code or with of the external implementations from Task 5.

2.4.3 Performance evaluation

The evaluation will be based on two types of outputs. Firstly, the assessment of the optical flow will be based on the Mean Magnitude Error (MMEN) of the estimated motion vectors and the Percentage of Erroneous Pixels (PEPN). The video stabilization will be assess when applied to the foreground extraction problem, so the precision, recall and F1-scores will be used for this task.

2.4.4 Deliverable

Progress slides: A *short* power point presentation (∼5 min.) commenting the used approach, results obtained and conclusions.

Source code: A working version of the Matlab code developed along the week. The code must be clearly commented and a README file must be included with instructions of how to use it.

Data: For the mandatory task, the estimated motion vectors in a plain text file and the obtained squared error, for each test sequence as well as averaged.

Data: The compensated sequences.

2.5 Week 5. Region tracking

The goal of this assignment is the implementation of a tracking system to uniquely identify the objects in the scene. This tracking will allow counting the amount of vehicles and obtain an approximate estimation of their speed.

2.5.1 Mandatory tasks

Task 1 Use a Kalman filter to track each vehicle appearing in the HIGHWAY and TRAFFIC scenes. Apply the background substraction work previously developed.

Task 2 Detect and labelling each car with a bounding box and a unique numerical identifier. The car ID must correspond to the object counter.

Task 3 Estimate the speed of the vehicles by formulating any simple assumption that will simplify the computation.

2.5.2 Optional tasks

Task 4 Test another object tracking method and compare the results with the Kalman filter. Provide the source code and references in your submission.

Task 5 Record your own video sequence of traffic monitoring, choosing a viewpoint, road geometry and static or handheld video. Process it with the available tools.

2.5.3 Performance evaluation

The location of the bounding boxes will be assessed according to the criterion defined in the Pascal Visual Object Challenge, which states that a detection is correct of there is an overlap of more than 50% with the ground truth box. In the case of the optional tasks, performance will be based on the delivered video demos.

2.5.4 Deliverable

- Progress slides: A *short* power point presentation (∼5 min.) commenting the used approach, results obtained and conclusions.
- Source code: A working version of the Matlab code developed along the week. The code must be clearly commented and a README file must be included with instructions of how to use it.
- Data: For the mandatory task, the estimated motion vectors in a plain text file and the obtained squared error, for each test sequence as well as averaged.
- Video: The resulting videos must be delivered clearly showing the obtained results.

3

Material

3.1 Programming language

The whole project will be developed with Matlab.

The recommended development environment is the Matlab platform, but any text editor can be used. Documentation on using Matlab may be found at MathWorks webpage.

We will use the Image Processing Toolbox and, not mandatory but helpful in some tasks, the Computer Vision Toolbox.

Other interesting resources are:

- Matlab general tutorials.
- Matlab tutorial in spanish.

3.2 Required libraries

For the development of the project, the Image Processing Toolbook is necessary. The Computer Vision Toolbox is not needed but may be recommended for some tasks.

3.3 Code

The following Matlab functions are available for this project:

- An alternative implementation of the Staunffer and Grimson approach.

3.4 Virtual machine

A virtual machine with Matlab installed and the dataset is available for the students. To connect to the server, you need the following data:

- SSH: 147.83.91.181:2241
- X2go, download client from x2go page [`http://wiki.x2go.org/doku.php/start`]. Configure as:
  - host: 147.83.91.181
  - login: ihcv$X$(X is the group number as 01,02,03,..)
  - password: ihcv (change it to suit your needs the first time you enter)
  - SSH port: 2241
  - Session type: GNOME

3.5 Datasets

In this project we use two public datasets, the Change Detection Video Dataset and the KITTI Vision Benchmark Suite.

3.5.1 Change Detection Video Dataset

The Change Detection Video Dataset is composed of indoor and outdoor visual data captured in surveillance and smart environment scenarios. The dataset includes a set of human-annotated ground truth of the change/motion areas to enable a precise quantitative comparison and ranking of various segmentation algorithms.

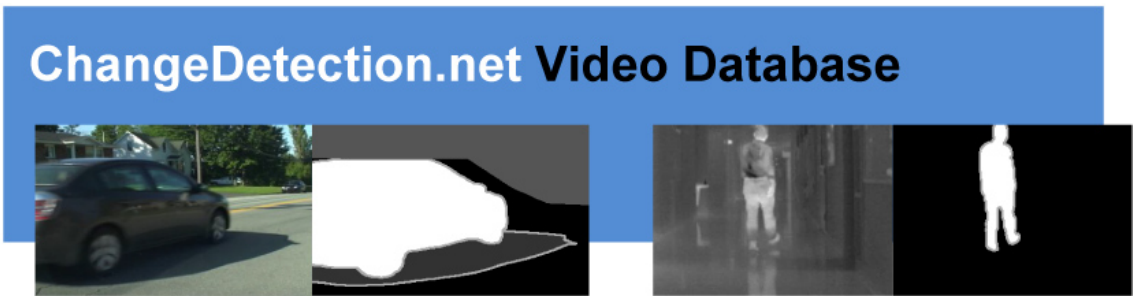


Figure 3.1: Change Detection dataset

Make sure to read carefully the *DATASET* section to understand the format of the ground truth provided. We will use a small subset (detailed in each of the weekly assignments) of this dataset.

3.5.2 KITTI Vision Benchmark Suite

The KITTI Vision Benchmark Suite is a dataset captured by driving a car around the city of Karlsruhe. It provides color video sequences in raw data format together with optical flow ground truth for each pixel. Please note that the video sequences provided by the KITII database are recorded using a stereo camera so 2 views for each sequence are given (*image_0* and *image_1*) and they are only of two (2) frames duration. We will always use one single view in these project (*image_0*).

Figure 3.2: Car used to capture the KITTI dataset

Make sure to download the stereo/optical flow development kit (1 MB) in the *FLOW* section and read the provided *readme.txt* file to understand how the optical flow ground truth is stored. They also provide functions to read it in C/C++ and Matlab.

This dataset will be used to evaluate the optical flow estimation implemented for video stabilization. We will use a small subset of the dataset detailed in the weekly assignments.

3.6 Reference books

1. "Computer Vision: Algorithms and Applications", Richard Szeliski. [`http://szeliski.org/Book/`].

2. "Computer Vision: A modern approach", David Forsyth and Jean Ponce. [`http://luthuli.cs.uiuc.edu/ daf/book/book.html`].

3. "Applied Signal Processing. A Matlab-Based Proof of Concept", Thierry Dutoit and Ferran Marqués, Springer US, 2009.

4

Evaluation

The mark V is assigned based on the evaluation of the following three parts.

Project Development (PD) The project will be developed by groups of students (3-4 students). The project is organized in five weeks, where the last lecture (sixth week) is devoted to project presentation and final evaluation. The task of each week is detailed in section 2. As a result of the weekly development of the assignment, each group has to deliver:

Code (Cd) : a working version of the software code developed along the week;

Weekly update (Wu) : a short set of slides delivered in PDF summarizing (1) used approach, (2) results, (3) problems and comments.

Both Cd and Wu have to be submitted to the digital campus **24 hours before** the monitoring lecture. Missing this deliverable will correspond to a zero mark for that week.

The code Cd, presentations Wu and feedback to project professor of the first 4 weeks PD_i^{week} will correspond to the $4 \times 15\%$ of the final mark.

For week 5 the short presentation is changed for a report, including the previous weeks 1 to 4 and the development performed in week 5. This report will correspond to the 30% of the final mark, PD_5^{week}.

Project Presentation (PP) The final project presentation will be evaluated. The PP mark will be composed of a mark evaluated by the professor and by the other students in the course and it will correspond to 10% of the final mark:

$$PP = 0.5 \cdot PP^{professor} + 0.5 \cdot PP^{students}$$

Intra-Group Evaluation (IGE) An internal evaluation is also foreseen for each group. This evaluation will be done by each student in the group in order to fairly evaluate the contribution of each group's member. The result of this evaluation will create a factor α_{IGE} that will modify the final mark of the project.

According to the described criteria, the final score V will be assigned as:

$$V = \sum_{i=1}^{4} 0.15 \cdot PD_i + 0.3 \cdot PD_5 + 0.1 \cdot PP$$