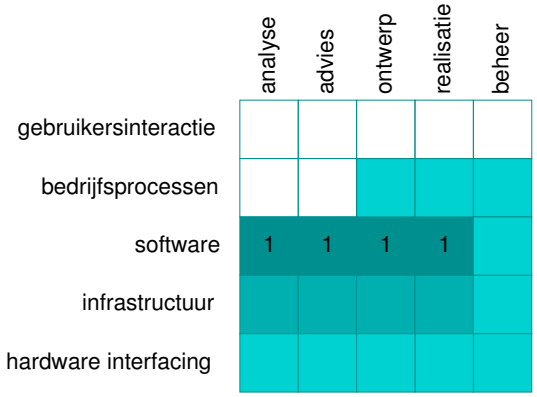
HOGESCHOOL ROTTERDAM / CMI

Augmented Reality

CMIAAR01K

ECTS: 2
Module responsible: Francesco Di Giacomo

Module description

Module name:	Augmented Reality
Module code:	CMIAAR01K
Number of ECTS and number of individual study hours:	This module gives 2ECTS, which corresponds to 56 hours. • 8 × 30 minutes frontal lecture • 8 × 90 minutes practicum • 8 × 300 minutes individual study
Examination:	Practical assignments and practical assessment
Course structure:	Lectures and practicums
Required knowledge:	Object Oriented programming.
Learning tools:	• Since the course covers only basic topics, we will use Wikipedia as a basic reference for the algorithms. • Vendor websites for documentation on tools and libraries. • Text editors: Visual Studio, NetBeans, IntelliJ, etc.
Connects to competencies :	

l check

Overall learning goal:	The student is able to implement basic image processing algorithms, which are the foundation of augmented reality.
Detailed learning goals and topics:	• The student is able to implement a RGB to YUV conversion, and YUV to RGB conversion; • The student is able to implement a greyscale filter; • The student is able to implement basic interpolation techniques applied to image rescaling. • The student is able to implement color clustering using the K-Means algorithm; • The student is able to implement the omnidirectional camera image unwarping; • The student is able to communicate in correct Dutch or English, using the correct jargon, about image processing and augmented reality.
Programming languages allowed:	• C# • Java • Python • F# • Scala • Haskell
Module responsible:	Francesco Di Giacomo
Date:	10-12-2015

1 Learning goals

The overall goal of the course is to provide a detailed answer to the questions:

- what is augmented reality?
- How do augmented reality algorithms work?

In this section we discuss further the full breadth of what the course covers, plus the desired level of skills achieved by the students.

1.1 Introduction

Augmented reality is the view of the physical world augmented by computer-generated visual or sound enhancements. It differs from virtual reality because the latter replaces the representation of the physical world, rather than augmenting it.

A wide number of problems of augmented reality is solved using image processing and computer vision algorithms. Common problems in this scope are: (*i*) image filtering, (*ii*) image clustering, (*iii*) edge detection. The solution of these problems is used in a wide range of applications such as surveillance, traffic control, military devices, and healthcare.

Through the knowledge acquired in this course students will be able to understand and implement the most common algorithms used for image processing. In this way they will be able to apply their knowledge to process the input of visual devices such as webcams, surveillance cameras, IR-images, etc.

1.2 Learning tools

Obligatory:

- Presentations and sources presented during lectures (found on N@tschool);
- Assignments to work on during practicums (found on N@tschool);
- Text editors: Visual Studio, NetBeans, IntelliJ.

2 Course program

The following is a comprehensive and detailed list of the course program:

1. Colour space encodings, basic image filtering.
    - RGB encoding.
    - YUV encoding.
    - Conversion between RGB and YUV encodings.
    - Grayscale filter with RGB and YUV encodings.
2. Interpolation applied to image resizing.
    - General idea behind interpolation.
    - Nearest neighbour.
    - Linear interpolation.
    - Bilinear interpolation.
    - Image resizing.
    - Interpolation sampling for image resizing.
3. Clustering and Image segmentation.
    - Definition of *Cluster*.
    - The clustering problem.
    - K-Means algorithm.
    - Application to image segmentation.
4. Omnidirectional cameras
    - Omnidirectional cameras.
    - Polar to Cartesian coordinates conversion.
    - Unwarping algorithm.

3 Testing and evaluation

In this section we discuss the testing procedure of this course, and the grading criteria.

3.1 Overall description

This module is tested with a series of practical assignments and a practical assessment. The assignments can be found on N@tschool.
At the practical assessment the teacher reserves the right to ask any question about the assignments and the student will be asked to complete simple exercises concerning the course topics.
Foreword and notes:

- The final grade is determined by the score in the assignments and the score of the practical assessment.
- The assignments ask to implement the computer vision algorithms seen in class.
- The assignments are individual.

This manner of examination is chosen for the following reasons:

- By implementing the algorithms seen in class the students can understand better the theoretical aspects of the topics and gain experience in using image libraries for the main programming languages used in the industry. (learning goals *image manipulation, computer vision algorithms*).
- At the practical assessment students must communicate about their code and prove their knowledge on the topics of the course. (learning goal *communication*).

The final grade is a weighted mean computed as follows:

- Assignments (40%)
- Practical assessment (60%)

The final grade can be computed as: `finalGrade = 0.4 * assignmentsGrade + 0.6 * practicalAssessment`.
In order to pass the exam, all the parts must be sufficient (i.e. partial score ≥ 5.5).

3.2 Assignments

3.2.0.1 Assignment 1 - Colour spaces and B&W filter Students must implement a function which converts an image with RGB encoding to YUV encoding, and a function which converts an image with YUV encoding to RGB encoding. Students must also implement the B&W filter algorithm in RGB space and YUV space.

3.2.0.2 Assignment 2 - Image resizing Students must implement a program to resize an image using both the nearest neighbour and bilinear interpolation techniques seen in class.

3.2.0.3 Assignment 3 - Clustering and image segmentation Students must implement the K-Means algorithm and apply it to find the colour clusters of an image. They must implement both a naive version and one using image resizing.

3.2.0.4 Assignment 4 - Omnidirectional cameras Students must implement the image unwarping algorithm for omnidirectional cameras.
You can find a detailed description of the assignments on Natschool.

3.3 Grades

Assignments 1 ,2, and 3 grant the student to get a 7 in the assignment score. Assignment 4 is optional and allows the student to score up to 10. The score starts from 1.

Assignments:	**Value in grades**
1	1
2	2
3	3
4	3

The assignments must be delivered the day before the practical assessment. The hand-in of the assignments is made of the source code of the program (or the Visual Studio template in case of C#) on N@tschool, and the report.

3.4 Feedback

It is possible to discuss the assignments (and their evaluation) during the practicum lectures or on appointment.