



Evaluating and Grading Students in Large-Scale Image Processing Courses

Nicole M. Artner, Ines Janusch, and Walter G. Kropatsch
TU Wien

In undergraduate practical courses, it is common to work with groups of 100 or more students. These large-scale courses bring their own challenges. Typically, students are evaluated based on regular hand-ins, exercise interviews, and/or tests. The problems solved by the students for the hand-ins are meant to give them a deeper understanding of a particular topic (such as edge filters). Unfortunately, these problems are in general too small and lack “the big picture,” so the students never experience what it means to solve a “real” problem. Furthermore, it is tiring and boring for the teaching staff to grade the same problem 100 times. Last but not least, it is nearly impossible to detect cheating because there are only a limited number of valid solutions to the problem.

Based on our experience with a traditional large-scale practical course in image processing, we developed a novel course approach based on the following goals:

- Increase student motivation by allowing them to solve their own projects.
- Let the students experience different aspects of a “real” research project.
- Show the user's view to the students.
- Evaluate and grade the results of the students in an objective manner.
- Keep the workload of the teaching staff manageable.

EDBV

The result of our efforts is a new approach to teaching “Introduction to Digital Image Processing” (or EDBV, from the German course title *Einführung in die Digitale Bild-Verarbeitung*).

EDBV is a mandatory, one term, practical course for all undergraduate students of media informat-

ics and visual computing and medical informatics at the TU Wien during their third term, offering three European Credit Transfer System (ECTS) credit points for 75 hours of workload. The course builds on the basics taught in the second-term visual computing course. Hence, most students in EDBV already have some background knowledge in image processing. Nevertheless, our course is also successfully completed by students from other specializations (such as mathematics and software engineering).

Since the winter 2012 term, EDBV has been taught in two phases: projects solved in groups and evaluation of projects. The phases of our image processing course not only aim to help the students memorize and absorb the course content, but they also cover all learning targets shown in Figure 1.¹ Table 1 gives an overview of the milestones of the whole course.

For communication, organization, and evaluation we rely on the online learning platform Moodle (<https://moodle.org>).

Phase 1: Group Projects

In the first two thirds of the course, groups of four to five students work on different projects. This phase begins with a challenging problem for the students: they have to specify their own problem to be solved within their project. To guarantee that each group works on a unique project, the problems are made public with the title, main goal, and a data sample (in most cases an image) in Moodle, via a first-come first-serve principle. As soon as a project is reserved in Moodle, its topic is no longer open to other groups.

The outcome of this first step in phase 1 are specifications of the projects. They include a problem statement, a proposal for the methodology to solve the problem, requirements and conditions

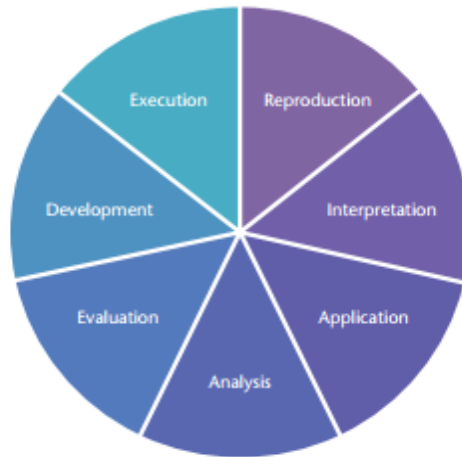


Figure 1. Learning targets based on a revision of Bloom's taxonomy.¹ The EDBV course uses a two-phase approach of projects and student/group evaluations to meet all targets.

Table 1. Example schedule of our course (winter term 2014).

Lesson date	Lesson title
01-10-2014	Introduction to the course, beginning of group building
10-10-2014	Start: Reservations of problems
19-10-2014	Deadline: Group building and hand-in of project draft
02-12-2014	Intermediate presentations
07-01-2015	Deadline: Hand-in phase 1; project results
12-01-2015	Deadline: Selection of evaluation project for phase 2
28-01-2015	Final presentations
30-01-2015	Deadline: Hand-in phase 2; evaluation

for the input data, a data sample, and an evaluation plan. This task is daunting for students in their third term. Therefore, we help them by providing a collection of recommended methodologies, support via discussion boards in Moodle, and personal contact with tutors. The recommended methodology includes standard image processing techniques, such as example edge and blur filters, Hough transformation, distance transformation, mathematical morphology, connected component labeling and region growing, as well as more challenging algorithms like SIFT (scale invariant feature transform) or optical flow. We recommend the introductory textbooks by Sonka² and Burger³ for further reading regarding the methodologies.

Based on this draft, each group receives feedback from us. We focus on the following questions: Is this a feasible project (regarding the duration and previous knowledge)? Do the chosen methods fit the defined constraints and goals? Are further conditions and requirements needed? Is

a reasonable and thorough evaluation planned? Putting together the draft and choosing/describing the dataset already covers the reproduction, interpretation, application, and analysis learning goals shown in Figure 1.

At this stage, each group is assigned to a tutor, who will guide them during the implementation of their project. Furthermore, the tutors build a bridge between teachers and students. They keep us up to date about the groups' progress and inform us if there are urgent or difficult problems, where they need our intervention.

All groups implement their projects in Matlab (www.mathworks.com) because it is a convenient environment for prototyping. Furthermore, phase 2 becomes much easier if all projects are coded in the same environment. We recommend that our students first implement their projects by employing the Matlab image processing toolbox, which mostly covers all the functionality they need to solve their problems. However, for the final prototype, the chosen image processing methods and algorithms should be implemented by the students themselves. Here we meet the development and execution learning targets from Figure 1.

At the end of the phase 1, the groups have to hand in their prototype, the assembled dataset, and a project report. This report should explain the methodology in theory, describe the group organization, give relevant implementation details, and present the evaluation. We ask the students to think of their colleagues as the target group of the report and to write it accordingly. Preparing the report covers several of the mentioned learning targets, where evaluation plays the most important role. In this way, all learning targets are fulfilled in phase 1 of the EDBV course, and then we revisit each during phase 2.

Continuous Supervision

Phase 1 requires continuous supervision of the students to make sure that they can achieve their goals and do not get stuck in minor details or fail as a result of some unanticipated problem. At least one meeting each month with the assigned tutor is mandatory until the phase 1 deadline. The aim of these meetings is to discuss the current state of the project and identify possible open questions and problems. Each tutor oversees up to six groups (up to 30 students) per term.

In addition to the tutor supervision and continuous, online support in Moodle (discussion boards), there are two oral presentations of the projects in front of the teaching staff and all the EDBV students. The first presentation is in the

middle of phase 1. It gives the teaching staff the opportunity to get an idea of the progress of the work, detect possible future problems, and advise on open problems. During the presentations, the students are also given an overview of all other ongoing projects and the employed methodologies. This gives every student a basic knowledge about all used methods and their applications, even if they are not using a particular method in their project. Furthermore, the first presentation helps the students to choose a project for phase 2. The final presentations show the groups' solutions of the previously defined problem. Nevertheless, it is also acceptable to present a methodology that did not successfully solve the problem and explain why (under the condition that a sound evaluation was done).

Phase 2: Evaluation of Projects

Phase 2 is essential for the proposed course mode. It tackles several problems:

- It is difficult to objectively evaluate and grade group projects because they cannot easily be compared with each other.
- It is not possible to judge the performance of single students.
- An objective evaluation and grading would generate an immense workload for the teaching staff.
- The student groups are never confronted with the weaknesses and bugs that survived their self-evaluation.

The basic idea of phase 2 is to involve the students in the evaluation process of the projects. Every student evaluates another group's project. All members of a group (during phase 1) need to evaluate different projects in phase 2, and of course nobody should evaluate their own project. For this evaluation, we provide the students with a standardized evaluation form. They have to discuss the results of their chosen project based on a set of questions, and they are asked to distribute a certain number of points. Points are given for concise and understandable explanations, sound methodology, thorough evaluation, and so on. In addition, they need to run the code and verify the results of the project report's evaluation. We also ask them to produce a small dataset and evaluate if they can achieve comparable results with the new data. Often, the dataset provided by the project group is biased and the results with the new data produced for phase 2 are worse than those presented in the final report.

In the end, we not only have the evaluation

Table 2. Course statistics.

Term	Students	Projects	Tutors
Winter 2012	103	22	5 (3.2 hours/week)
Winter 2013	114	24	5 (4 hours/week)
Winter 2014	83	17	5 (4 hours/week)

of the projects by the assigned tutors and teachers, but also several completed by other students. Because the teaching staff uses the same evaluation forms as the students, the combination and comparison of evaluation results is straightforward. Individual checking can focus on diverging evaluations and outliers (for example, a student might have found a bug that others overlooked). The final grades are given exclusively by the teachers. However, the grading decision for the group projects is clear if all student and tutor evaluations coincide. In case of discrepancies, the evaluation reports are checked and the evaluation results are weighted according to the arguments provided in the reports. The hand-ins of phase 2 are graded according to comparison and agreement with other evaluations of the same project and the quality of the arguments given to justify the assigned points.

Statistics and Course Details

The EDBV course has been held three times (every winter term) since 2012. The teaching staff always consisted of one professor, one university assistant, and five tutors. Table 2 gives brief course statistics.

For the sake of convenience, the maximum number of grading points for EDBV has been set to 100. These are split into 80 points for phase 1 and 20 points for phase 2. This point distribution only allows excellent students, who excel in the group and the single work, to achieve the best grade (which starts with 88 points).

Student Results

In total, 63 projects have been completed by student groups so far. The following list shows a small excerpt of previous project topics:

- Image stitching (see Figure 2)
- Color by numbers (see Figure 3)
- Perspective rectification
- QR-code/barcode reader
- Matching pairs of a memory game
- Counting coins/numbers on dice
- Reading sheets of music
- Solving mazes
- Face morphing

For this article, we selected two outstanding projects. Project 1 is about image stitching. Its

Education

Figure 2. Example project 1. Image stitching student project (a) input and (b) output.

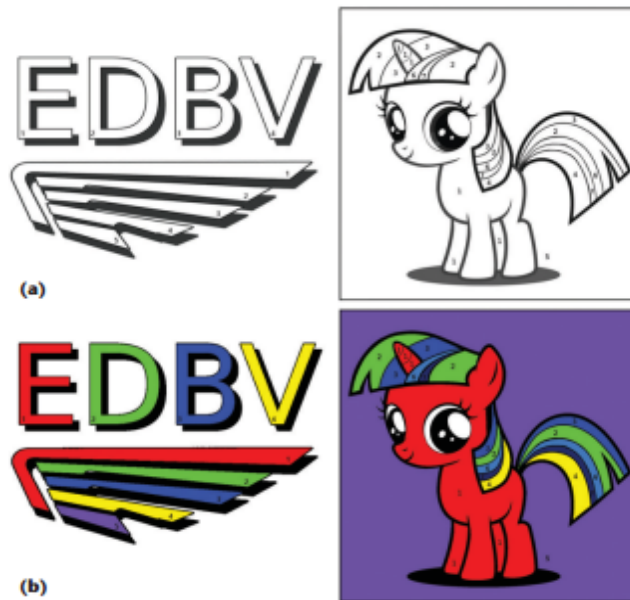


Figure 3. Example project 2. Color by numbers student project (a) input and (b) output.

aim was to stitch two images together, where the images show the same scene and contain overlapping image content (see Figure 2). For this project, the students implemented the SIFT algorithm⁴ to detect corresponding interest points in the two images. The stitching was done using multiresolution splines.⁵

Project 2 is about *Malen nach Zahlen* (or "color

by numbers" in English). Here the aim was to fill out an outline image based on numbers given in the connected components (see Figure 3). The individual regions were determined using connected component labeling. To identify the numbers, the following algorithms were tested and used: template matching, k-nearest neighbors, and neural networks.⁶ In a last step, the regions were filled in the colors selected by the user for a certain number.

This flexible course approach allows for continuous adaption and improvement. Therefore, we may change the presented mode slightly based on our experience in a previous term or based on feedback.

For example, for the 2014/2015 winter term, we added an alternative mode with fixed projects for phase 1. This mode offers predefined project drafts: problem statements, together with suitable methodology, datasets, and evaluation plans provided. The problems and the first project specifications demand a lot of creativity, research, and planning. For less dedicated students, the alternative mode provides a simplified option, of course at the expense of an inferior grade.

Moreover, our course mode allows us to offer additional activities. In January 2014, EDBV students had the opportunity to visit the University of Applied Sciences Upper Austria in Hagenberg and present their project results to students of similar courses there.

An attractive feature of such a two-phase course is that it allows for collaboration with another

university. The idea would be to exchange the results of phase 1 and have the evaluation of phase 2 done by the students of the other university. Even more universities could be involved, relaxing the constraint that the bilateral exchange of student groups should have approximately equal size.

If our course mode is interesting to you and you could imagine collaborating with us, do not hesitate to contact us. We will gladly share the necessary course materials. ❖

References

1. L.W. Anderson and D.R. Krathwohl, eds., *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*, Addison Wesley Longmann, 2001.
2. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, Cengage Learning, 2014.
3. W. Burger and M.J. Burge, *Digital Image Processing: An Algorithmic Introduction Using Java*, Springer Science & Business Media, 2009.
4. D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, 2004, pp. 91-110.

5. P.J. Burt and E.H. Adelson, "A Multiresolution Spline with Application to Image Mosaics," *ACM Trans. Graphics (TOG)*, vol. 2, no. 4, 1983, pp. 217-236.
6. Ø.D. Trier, A.K. Jain, and T. Taxt, "Feature Extraction Methods for Character Recognition: A Survey," *Pattern Recognition*, vol. 29, no. 4, 1996, pp. 641-662.

Nicole M. Artner is a university assistant in the Image Processing and Pattern Recognition (PRIP) group at TU Wien. Contact her at artner@prip.tuwien.ac.at.

Ines Janusch is a university assistant in the Image Processing and Pattern Recognition (PRIP) group at TU Wien. Contact her at ines@prip.tuwien.ac.at.

Walter G. Kropatsch is a professor and the leader of the Image Processing and Pattern Recognition (PRIP) group at TU Wien. Contact him at krw@prip.tuwien.ac.at.

Contact department editor Gitta Domik at domik@uni-paderborn.de and department editor Scott Owen at sowen@ysu.edu.

cn Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

Software Engineering Institute | Carnegie Mellon University

IEEE Computer Society | Software Engineering Institute

Watts S. Humphrey Software Process Achievement Award

Nomination Deadline: October 15, 2015

Do you know a person or team that deserves recognition for their process-improvement activities?

The IEEE Computer Society/Software Engineering Institute Watts S. Humphrey Software Process Achievement Award is presented to recognize outstanding achievements in improving the ability of an organization to create and evolve software.

The award may be presented to an individual or a group, and the achievements can be the result of any type of process improvement activity.

To nominate an individual or group for a Humphrey SPA Award, please visit <http://www.computer.org/web/awards/umphrey-spa>