# Integrating Mobile Robotics and Vision with Undergraduate Computer Science

Grzegorz Cielniak, Nicola Bellotto and Tom Duckett

*Abstract*—This paper describes the integration of robotics education into an undergraduate Computer Science curriculum. The proposed approach delivers mobile robotics as well as covering the closely related field of Computer Vision, and is directly linked to the research conducted at the authors' institution. The paper describes the most relevant details of the module content and assessment strategy, paying particular attention to the practical sessions using Rovio mobile robots. The specific choices are discussed that were made with regard to the mobile platform, software libraries and lab environment. The paper also presents a detailed qualitative and quantitative analysis of student results, including the correlation between student engagement and performance, and discusses the outcomes of this experience.

*Index Terms*—Robotics education, robot vision, robot programming, Student as Producer.

## I. INTRODUCTION

This paper describes the integration of robotics education into the undergraduate curriculum at the School of Computer Science, University of Lincoln, UK. Rather than teaching robotics in isolation, the subject is tightly integrated with other disciplines, in particular computer vision, as well as the ongoing research activities in the School. The proposed approach to robotics teaching features a holistic combination of theory and practical work, including the programming of vision-guided mobile robots in an open-ended assignment loosely based on the popular RoboCup football tournament. This strategy also reflects the current policies of the institution on research-informed teaching and the Student as Producer [1], a university-wide initiative supported by the UK Higher Education Academy.

Student as Producer "restates the meaning and purpose of higher education by reconnecting the core activities of universities, i.e., research and teaching, in a way that consolidates and substantiates the values of academic life" [1]. It does not dictate strict rules or policies but rather acts as a framework that encourages thinking about educational processes with the student engagement and active participation in mind. The core idea of the initiative is rooted in constructivism, which also inspired Papert's constructionism [2]; it was Papert who advocated the use of technology in the learning process. This direct link makes Student as Producer an especially attractive framework in the current context, inspiring some of the choices made and the decision to involve students at various stages of preparation of the Robotics module (e.g., the selection

G. Cielniak, N. Bellotto and T. Duckett are with the School of Computer Science, University of Lincoln, UK. e-mail: {gcielniak,nbellotto, tduckett}@lincoln.ac.uk

of software and hardware platform, creation of assessment scenarios, etc.).

This paper presents the experience gained from conducting these activities, the choices made during preparation of the theoretical and practical work, the design of assignments, and various ways of assessing student performance.

## II. RELATED WORK

In recent years, the use of robotics in education has gained a significant interest among robotics experts and educators, illustrated by a number of funded projects (e.g., TereCop [3], Roberta [4]), workshops and conferences (e.g., [5], [6]). Due to its interdisciplinary nature, robotics is often delivered in combination with other complementary subjects like electronics, mechanical engineering, computer vision [7], and the like. However, robotics is not only being taught as a specialist subject to a narrow audience of future robotics specialists [8], but is being increasingly used as a tool for improving knowledge of other subjects, developing transferable skills, enhancing student engagement in science, and more. This is closely related to the recent increase in the availability of affordable robotics platforms but is also due to the potential attractiveness of robotics technology to younger generations.

Previous work in educational robotics reports teaching activities at different educational stages, including the primary, secondary [9] and tertiary level [10]. These activities, depending on the level, address various learning aspects including manual and communication skills [11], and scientific methods such as measurement, calculations, problem-solving, etc. Robotics is also being used to improve understanding of other subjects like maths, physics and engineering (e.g., [12]) and in general for improving student engagement in science subjects, [4]. However, the efficient use of robotics tools requires specialist training for teachers themselves, especially those involved in the early stages of education. While for more technically-oriented teachers this comes naturally, many others feel intimidated by the complexity and the practical skills required [13]. Educational robotics is also being used to address social issues including gender balance, disabilities or disadvantaged social backgrounds (e.g., [11], [14]).

While there are many comprehensive sources covering theoretical aspects of robotics (e.g., [15], [16], [17]), the main challenge lies in the preparation of appropriate practical activities. The main difficulty comes from the fact that both pedagogical (e.g., specific age, or learning objectives) and technical aspects have to be considered. A key decision related to the latter is the choice of robotic platforms, taking into account reliability,

| Lecture | Topic | C1 | C2 | C3 |
|---------|-------|----|----|----|
| A1 | Intro. to Computer Vision | ○ | | ○ |
| A2 | Intro. to Linear Algebra and MATLAB | ● | | |
| A3 | An Overview of Pattern Recognition | ● | | ○ |
| A4 | Spatial Processing and Filtering | ● | ○ | ○ |
| A5 | Color Image Processing | ● | ○ | ○ |
| A6 | Morphological Image Processing | ○ | ○ | ● |
| A7 | Image Segmentation I | ○ | ○ | ● |
| A8 | Image Segmentation II | ○ | ○ | ● |
| A9 | Image Representation and Description | | | ● |
| A10 | Pattern Classification | | | ● |
| B1 | Introduction to Robotics | | ○ | ○ |
| B2 | Robot Programming in C# | | ● | |
| B3 | Actuators and Sensors | | ● | ● |
| B4 | Robot Vision | | ● | ○ |
| B5 | Robot Control | | ● | ● |
| B6 | Robot Behaviors | | ● | ● |
| B7 | Control Architectures I | | ○ | ● |
| B8 | Control Architectures II | | ○ | ● |
| B9 | Navigation Strategies | | | ● |
| B10 | Robotic Map Building | | | ● |

TABLE I
MAIN TOPICS IN SEMESTER A (COMPUTER VISION) AND SEMESTER B
(MOBILE ROBOTICS), WITH INDICATIVE COVERAGE BY ASSESSMENT
(● = HIGH, ○ = LOW), DETAILED IN TABLE II.

| Assessment | Weighting | Description | Semester |
|------------|-----------|-------------|----------|
| C1 | 30% | Image Processing in MATLAB | A |
| C2 | 30% | Vision-based Robot Control | B |
| C3 | 40% | Written Examination | A + B |

TABLE II
THE THREE ASSESSMENTS FOR THE MODULE INCLUDING WEIGHTINGS.



Fig. 1. The layout of the computer lab used for practical sessions (left) and a snapshot taken during one of the sessions (right).

ease of use, maintenance, and so on. Thanks to the increasing number of affordable robots available on the market, this choice is somewhat easier (see [18] for an extensive review of the available robotics platforms). In addition, affordable robot kits have become popular toys and gadgets, further stimulating interest in robotics among younger generations. There are several popular scenarios that are employed for practical activities including robotic football, home services, rescue missions [19], [20], or Braitenberg vehicles [21]. The scenarios are acting in this case as micro-worlds [2] in which there are well understood objectives and requirements, as well as potential for experimentation and discovery.

### III. MODULE DESCRIPTION

Robotics at the Lincoln School of Computer Science is delivered primarily in a study module called Computer Vision and Robotics. The module is targeted at the third-year (i.e., final-year) undergraduate students due to its advanced content, its assumption of significant programming skills and its focus on mobile robots as complete systems, as well as on their components. The module is compulsory for students on the Computer Science program, and optional for students studying other programs at the School (Games Computing, Computer Information Systems, and Web Technology). There are several relevant modules that students undertake in earlier years of study which provide the necessary background in programming and basic knowledge of Artificial Intelligence (AI),
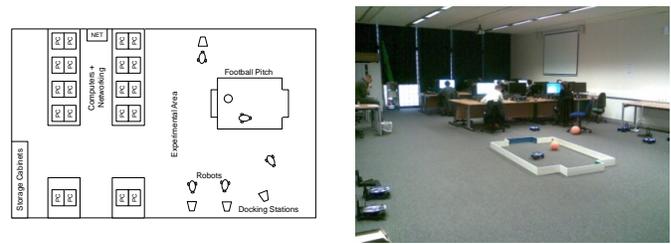
including Software Development in year 1, and Advanced Software Development and AI in year 2. In addition, students have the option to study robotics as part of their self-guided Individual Study Project in year 3, in parallel with this taught module.

Computer Vision and Robotics consists of two distinct parts: computer vision fundamentals are covered in the first semester (A) and robotics is covered in the second semester (B) – see Table I for a detailed list of topics, and Table II for their weightings. The computer vision part has a particular emphasis on pattern recognition applications, hence the early overview on this topic, but otherwise follows a fairly standard selection of topics in digital image processing from a well-known textbook [22]. The weekly one-hour lectures, with additional supporting materials, are accompanied by a weekly two-hour workshop in which the students learn to program in MATLAB using the Image Processing Toolbox (IPT) and the corresponding textbook [23]. Many techniques and concepts learned in the computer vision part of the module are directly relevant and applicable to the robotics content. Indeed, robot vision is a primary focus of the practical sessions and the assignment task in the second semester. Each semester is individually assessed through a practical assignment (see Section IV) and there is a final written examination covering the theoretical content of both semesters. The robotics part of the module includes 12 weeks of lecture delivery (comprising ten major topics plus two weeks for supporting activities) and practical workshop sessions. The topics covered include the main challenges of robotics, robotic components, relevant software libraries, robot vision and control, robotic architectures, navigation strategies and map building. The first set of lectures provides many practical examples related to the robotic platform used, so that the students can experience a direct link between theory and practice. The material covered also includes examples from research conducted in the School to illustrate fundamental problems of perception (e.g., detecting people) and control (e.g., navigation), though the focus is mainly on "textbook" science where possible [15], [16], [17].

### A. Practical Sessions

The robotics part of the module features 12 two-hour long workshop sessions where the students have access to the robotic equipment. The sessions take place in a dedicated computer lab with storage facilities and necessary space for experimentation with the robots, Fig. 1. With this arrangement, all frequent tasks such as unpacking, setting-up, charging, and
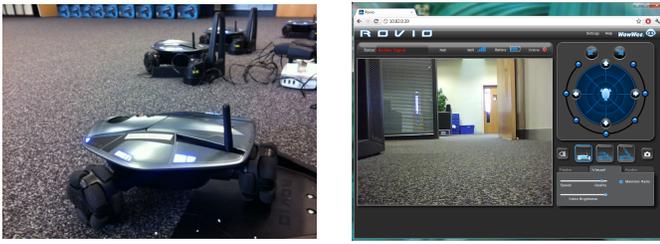
Fig. 2.   Rovio, the mobile robot (left) and the robot's web interface (right).

| Feature | Rovio | Roomba | NXT |
|---|---|---|---|
| Affordability | ● | ● | ★ |
| Maintenance/charging | ● | ★ | ○ |
| Set-up difficulty | ● | ★ | ★ |
| Documentation | ★ | ★ | ★ |
| Community Support | ★ | ○ | ● |
| Software components/libraries | ● | ● | ● |
| Quality of components | ★ | ★ | ● |
| Vision support | ● | ○ | ○ |

TABLE III
A COMPARISON OF ROBOTIC PLATFORMS USED FOR TEACHING ROBOTICS
IN THE CLASSROOM (SUPPORT: ● = GOOD, ★ = FAIR, ○ = POOR).

so on can be carried out efficiently and smoothly without limiting the amount of time for interaction and work with the robots. In addition, the students can access the lab outside the workshop hours, when it is not reserved for other activities. This arrangement appeared to be very popular in the most recent offering of the course, especially towards the assignment hand-in date (!). The first four workshop sessions are designed as introductory tutorials where the students learn about the basic components of a robot, relevant software libraries and programming principles. The tasks include students writing their own object detection algorithms, developing visual feedback controllers, implementing simple behaviors or using Finite State Machines for sequencing more complex behaviors. The tutorial tasks were designed to provide a solid knowledge base as well as preparing necessary components needed for developing the following assignment task (see Section IV-B). The preparation of these teaching materials was assisted by a Master's degree student who had studied the module in the previous academic year, as a ten-week summer project supported by the university in connection with the Student as Producer program [1]. The university's policy of research-informed teaching is also aided by the part-time employment of current Ph.D. students in the area of vision and robotics as demonstrators during the workshop sessions.

### B. Choice of Robot Platform

A number of choices had to be made regarding the platform and software libraries. Since the targeted students are exposed mostly to programming in C# in their previous years of study, it was decided to adopt this programming language for the practical sessions. The students are provided with a simple software wrapper directly implementing robot commands as specified by the Rovio API document. An alternative would be to rely on existing robotic suites such as Player/Stage, ROS or Microsoft Robotics Developer Studio; however, such a choice is perhaps more preferable for larger scale, long-term projects as it requires substantial effort and time to become familiar with these frameworks. At the same time, the students learn what to expect from a commercial platform in terms of software support and how to extend the functionality to meet their own needs. The wrapper library [24] was developed following object-oriented programming guidelines so that the students can apply the concepts learned in previous years to a real physical platform. The recommended image processing library is AForge [25], which provides a well structured and documented functionality including a variety of image filters and object detection algorithms. Ideally, OpenCV [26] would

be preferable due to its overall functionality, maturity and support. However, the existing C# wrappers are inefficient and rely on non-safe use of the language that could hinder the learning process.

Rovio by WowWee [27] is an affordable mobile platform equipped with a set of sensors including a color camera (640x480 pixel resolution, 30 fps, RGB) mounted on a moving head, odometry, infrared global navigation sensor and an infrared obstacle detector (see Fig. 2). The omni-directional drive configuration with omni-directional wheels enables holonomic movement in all directions. The communication with the robot is realized through wireless Ethernet; the on-board computer (ARM-based architecture) runs a web-server that accepts requests from and sends information to the remote PC, which can be programmed to realize different custom behaviors. There are several alternative robotic platforms that are popular in delivering robotics courses, including Roomba and LEGO NXT. Table III presents a subjective comparison of the popular robotic platforms based on the delivery team's experience of using them for teaching robotics in the previous year.

While Roomba and LEGO NXT's many attractive features (e.g. reconfigurability, direct control of wheels) have made these platforms very popular at other institutions (e.g., [28], [29]), there are several reasons that make Rovio an ideal platform for the delivery of robotics in the proposed context. First, the robot is equipped with a color camera that is directly accessible from the software and can be used as the primary source of sensory information in the workshop tasks. Other platforms (Roomba, NXT) require additional hardware components and solutions to enable video streaming. The platform is affordable – currently, it costs less than a single Roomba robot or a standard Lego NXT set. This makes it possible to purchase a large number of units that can be used independently by individual students in relatively large groups, which is very important for maintaining student engagement in the workshops. This also makes Rovio a more expendable platform, as a broken or faulty unit can be easily replaced. A simple and sturdy design, together with a recharging station, result in a straightforward and easily maintained set-up that minimizes the time overhead for the preparatory tasks required before each session. The robot features a convenient communication interface through wireless Ethernet, enabling easy setup in existing computer laboratories (there are no drivers required, and no additional dongles, connections or the like)

and a straightforward API that simplifies development of the students' own robot behaviors, with minimal dependencies on other software libraries. Each Rovio robot in the lab is assigned its own IP address, and can be accessed directly through an ordinary web browser (an important consideration when trying to work with large groups of students).

The frequent and intensive use of the Rovio robots also revealed some limitations that might have not been immediately obvious. Among them, the odometry sensor is unreliable, the global navigation sensor is unusable in a multi-robot environment, the camera image is of low quality, there can be connection and bandwidth problems which are partially caused by the existing network facilities, and so on. On the other hand, these limitations helped the students to understand some of the fundamental issues and trade-offs in robotics and other embedded systems, including networking, real-time control, sensor noise, interaction of multiple complex systems, etc.

## IV. Assessment

The assessment strategy involves three separate components. Details of the topics covered by these three assessments are shown in Table I. The main objective of the first and second components, "Digital Image Processing in MATLAB" (C1) and "Vision-based Robot Control" (C2), is to assess student practical work during Semester A and B, respectively; the marks for these two components mostly reflect the functionality of the developed systems. The last component (C3) is the final (closed-book) written exam, designed to assess student comprehension of theoretical topics from the whole academic year. Each component carries a weighting, Table II, which is used at the end of year to compute the overall module mark.

### A. Digital Image Processing in MATLAB (C1)

The scope of this component is to assess knowledge and understanding of various aspects of computer vision, in particular those related to image processing. This component is subdivided into three workshop tasks with increasing levels of difficulty, assessed at regular intervals throughout the semester, titled "Binary Images & Introduction to Pattern Recognition", "Intensity Transforms & Spatial Filtering" and "Color Image Processing & Face Detection". The deliverables for each task consist of a brief report describing the approach used to solve the problem and the results obtained, accompanied by the corresponding MATLAB source code. The latter is further demonstrated in a follow-up workshop session, where students are required to answer specific questions about their own MATLAB implementations.

### B. Vision-Based Robot Control (C2)

The aim of this assignment is to evaluate competence in two major learning outcomes: the application of computer vision techniques to solve practical problems, and the application of AI control methods to mobile robotics. The assignment, which builds upon computer vision expertise from the first semester and some knowledge of AI from the previous year, is inspired by the RoboCup competition [19].

The students are asked to design and develop a simplified version of robotic football. The students can select one type of player from a given list of striker, midfielder, defender and goal keeper. The game features a uniformly colored ball and a playing field consisting of a rectangular enclosure with distinctively colored goals installed in the computer lab, Fig. 1. The minimum requirement for player functionality includes a ball searching behavior and a striking/defending behavior (which could be implemented as a reactive steering behavior using a proportional controller), depending on the player type. Students are free to choose the vision and robot control algorithms. Extra credit is given for developing additional components, including, but not limited to: 1) an enhanced object detection system for learning object appearance or using multiple cues; 2) behavior coordination for deriving sophisticated game strategies and implementing awareness of other game objects, like goals and opponents; and 3) incorporation of the above information into the player behavior. The submission of this assignment includes a short technical report documenting the design, implementation and evaluation of the proposed functionality as well as the developed source code. In addition, it is compulsory for the students to demonstrate their work during a separate workshop session after the submission, so the technical achievement and originality of the submitted work can be assessed and graded.

### C. Exam (C3)

The exam covers theoretical topics from both parts of the module, and is particularly designed to evaluate students' critical assessment of the major topics covered. The exam is taken at a predetermined time and location, a few weeks after the end of Semester B, and must be completed within three hours. The students have to answer four questions, two from the three Computer Vision section questions, and two from the three Robotics section questions. There are no programming tasks in this final assessment. Instead, students are required to demonstrate their understanding of the systems and algorithms covered in the two semesters (see Table I for an overview), and to compare various solutions to computer vision and robotics problems.

## V. Student performance

In the academic year 2010/11, the student cohort consisted of 18 Computer Science students for whom the module was compulsory, four Games Computing students and one Web Technology student who chose the module as an option, giving 23 students in total. This is a relatively low sample and therefore the reported results are anecdotal to some extent and might not capture all characteristics of the course delivery.

### A. Overall Results

Fig. 3 presents the comparative distribution of marks for each assessment component, and sample correlation coefficients between marks for each pair of assessment components. While the practical sessions proved to be popular and the students received relatively good marks for both assignments (C1
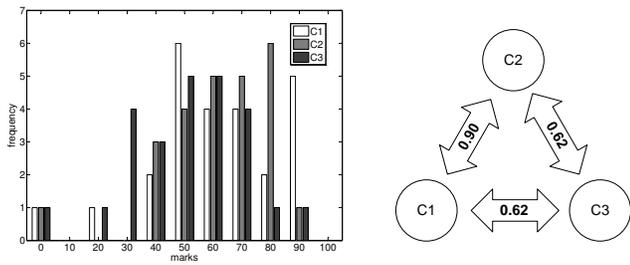
Fig. 3. Distribution of marks obtained for different assessment components (left) and mark correlation between each pair of assignment components (right).

| | CVR | ASE | ISP | ASD | AI | SD |
|---|---|---|---|---|---|---|
| SD | 0.34 | 0.41 | 0.38 | 0.37 | 0.53 | 1.00 |
| AI | 0.49 | 0.40 | 0.49 | 0.62 | 1.00 | |
| ASD | 0.62 | 0.54 | 0.54 | 1.00 | | |
| ISP | 0.87 | 0.87 | 1.00 | | | |
| ASE | 0.94 | 1.00 | | | | |
| CVR | 1.00 | | | | | |

TABLE V
SAMPLE CORRELATION COEFFICIENTS BETWEEN COMPUTER VISION AND ROBOTICS AND OTHER RELEVANT MODULES (SEE SECTION V-A FOR FULL LABEL DETAILS).

| | C1 | C2 | C3 |
|---|---|---|---|
| Lectures | 0.82 | 0.63 | 0.52 |
| Workshops | 0.70 | 0.55 | 0.37 |

TABLE IV
SAMPLE CORRELATION COEFFICIENTS BETWEEN THE MARKS OBTAINED FOR THE DIFFERENT COMPONENTS AND THE ATTENDANCE RECORDED FOR THE LECTURES AND WORKSHOP SESSIONS.

and C2), the theoretical examination (C3) results were lower than expected – see the low correlation coefficients between the exam and both practical assignments. These figures might indicate a better engagement of the students in the practical part of the course, but perhaps also a wider problem of student comprehension of theoretical material observed across all programs at the School.

To analyze the results further, attendance data was considered as a basic indicator of student engagement, Table IV. The results indicate that attendance at lectures and workshops showed a strong correlation with performance on the Computer Vision assignment (C1), although less so in the second semester. Perhaps surprisingly, the attendance data showed the lowest correlation with the marks obtained in the exam.

Further analysis included the correlation between marks obtained in Computer Vision and Robotics and other relevant modules taken by students during their course of study. Table V presents sample correlation coefficients in the form of a matrix between different modules including a first-year module, Software Development (SD); second-year modules, Advanced Software Development (ASD) and Artificial Intelligence (AI); and third-year modules Computer Vision and Robotics (CVR), Advanced Software Engineering (ASE) and Individual Study Project (ISP). While it can be seen that the correlation between CVR and other third-year modules is stronger than with modules from earlier years, it is interesting to note that CVR is ranked as the module most correlated with ASE, ISP and ASD, i.e., the modules where programming skills play a prominent role.

### B. Robotics Assignment – Observations

The robotics assignment had clearly defined minimum requirements but fairly open goals, which encouraged experimentation and exploration. This resulted in a number of exceptionally good submissions which included many features beyond the standard specifications. All the best submissions

(marks greater than 70%) had a clearly defined individual focus and explored different issues and directions. Some examples of the outstanding achievements presented by the students included:

- *object detection:* a multi-stage image processing pipeline including cascaded segmentation in different color spaces, morphological operators for noise filtering, the use of edges as additional features, and histogram-based tuning and learning of image filter parameters;
- *control architectures:* a hybrid architecture combining reactive and deliberative approaches, complex behavior sequencing models (e.g., hierarchical FSM), a predictive ball search behavior, multi-threading and synchronization featuring customized threading queue mechanisms and off-line system development using prerecorded data sets;
- *system evaluation:* rigorous quantitative evaluation including switching time analysis of behavioral models and dedicated testing scenarios, consideration of trade-offs (e.g., speed vs. accuracy) for robot controllers.

The above list of topics indicates that the students were able to apply not only knowledge learned in the previous semester (for example, object detection) but also that acquired in other modules including AI, Software Development and Software Engineering. Many of the listed techniques required the students to research sources other than the recommended reading material. On average, the students with the highest grade referenced two items from the recommended reading list, three items citing other books, journal publications and conference papers discovered through individual research, and one item citing software or hardware.

## VI. DISCUSSION AND CONCLUSIONS

It has been pointed out previously that the goal of "educational robotics in general, is not precisely to teach learners to be robotics experts but to develop the essential competences to be successful in the present world" [28]. The authors believe that the module presented here gives students vital expertise in areas that are otherwise not strongly covered in the "standard" computer science topics, such as dealing with complex systems at a systems (and systems of systems) level, combining hardware with sensing and control software, understanding the practicalities of real-time systems, understanding the inherent uncertainty in the real-world as perceived through sensors (sensor noise), and applying programming methodologies in practice. The students demonstrated a very high level of

engagement, spending a significant amount of time solving the robot football task required for the assignment. This resulted in a number of exceptional submissions with very original functionality that went beyond the assignment brief (incorporating such features as threading queues, speech synthesis, etc.). While many students struggled with the evaluation part of the assignment (most likely due to their time management), many of the technical issues and platform shortcomings were identified by the students (connection and bandwidth problems, granularity of the movement commands, limited odometry, changing light conditions, etc.). The laboratory space encouraged cooperation, support and competition in developing individual solutions. The module proved to be popular among the students and there was an increased intake (44 students) in the academic year 2011/12. Some pedagogical issues to be addressed in future extensions of this work include, for example, time requirements for solving practical assignments and scenarios alternative to RoboCup. Further development plans for the module include: a common software repository to teach code maintenance and development in teams, extensions to the software environment, multi-robot scenarios, and a greater involvement in the Student as Producer initiative [1], including recruiting more student helpers and building stronger links with the student Computing Society. A comparative study with the previous year and with other institutions delivering similar content is also envisaged.

## REFERENCES

[1] M. Neary and J. Winn, "The student as producer: reinventing the student experience in higher education," in *The future of higher education: policy, pedagogy and the student experience*. Continuum, 2009.

[2] S. Papert, "What is Logo? And who needs it?" in *Logo philosophy and implementation*. Logo Computer Systems Inc., 1999.

[3] D. Alimisis, M. Moro, J. Arlegui, A. Pina, S. Frangou, and K. Papanikolaou, "Robotics & Constructivism in Education: the TERECoP project," in *Proc. of the 11th European Logo Conf.*, Bratislava, Slovakia, 2007.

[4] Roberta Project. [Online]. Available: http://www.roberta-home.de/en

[5] International workshop: Teaching robotics teaching with robotics. [Online]. Available: http://www.terecop.eu/TRTWR2012.htm

[6] International robotics in education conference. [Online]. Available: http://www.rie2012.eu/

[7] G. Bebis, D. Egbert, and M. Shah, "Review of computer vision education," *IEEE Transactions on Education*, vol. 46, pp. 2–21, 2003.

[8] B. A. Maxwell and L. A. Meeden, "Integrating robotics research with undergraduate education," in *IEEE Intelligent Systems Magazine*, 2000.

[9] M. Kabátová, L. Jašková, P. Lecký, and V. Laššáková, "Robotic activities for visually impaired secondary school children," in *Proc. Int. Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum*, Riva del Garda, Italy, April 2012.

[10] J. Carter, S. Matthews, and S. Coupland, "Teaching robotics at the postgraduate level: Assessment and feedback for on site and distance learning," in *Proc. Int. Conf. on Robotics in Education*, 2011.

[11] T. Urschitz and T. Carl, "Mission on Mars. Interactive robotics session, on video conference, through the use of Interactive White Board (IWB)," in *Proc. Int. Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum*, Riva del Garda, Italy, 2012.

[12] L. Riano and M. McGinnity, "Design and validation of a robotic system to interactively teach geometry," in *Proc. Int. Conf. on Robotics in Education*. FEI STU, Slovakia, 2010, pp. 5–9.

[13] D. Alimisis, "Integrating robotics in science and technology teacher training curriculum," in *Proc. Int. Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum*, Riva del Garda, Italy, April 2012.

[14] D. Catlin and S. Robertson, "Using educational robots to enhance the performance of minority students," in *Proc. Int. Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum*, Riva del Garda, Italy, April 2012.

[15] G. Bekey, *Autonomous Robots: From Biological Inspiration To Implementation And Control*, ser. Intelligent Robotics and Autonomous Agents. Mit Press, 2005.

[16] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Bradford Company, 2004.

[17] R. R. Murphy, *Introduction to AI Robotics*, 1st ed. MIT Press, 2000.

[18] M. Ruzzenente, M. Koo, K. Nielsen, L. Grespan, and P. Fiorini, "A review of robotics kits for tertiary education," in *Proc. Int. Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum*, Riva del Garda, Italy, April 2012.

[19] M. Asada, R. D'Andrea, A. Birk, H. Kitano, and M. Veloso, "Robotics in edutainment," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 795–800.

[20] A. D. Tonkonogui, P. Bell, A. R. Linse, and A. V. Mamishev, "Integrating FIRST robotics program with university curriculum," in *Proc. Int. Conf. on Robotics and Applications*. Anaheim, CA, USA: ACTA Press, 2007.

[21] E. Datteri, L. Zecca, F. Laudisa, and M. Castiglioni, "Explaining robotic behaviors: a case study on science education," in *Proc. Int. Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum*, Riva del Garda, Italy, April 2012.

[22] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Prentice Hall, 2008.

[23] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB (2nd Edition)*. Gatesmark Publishing, 2009.

[24] RobotLib project. [Online]. Available: http://robotlib.codeplex.com

[25] AForge.NET. [Online]. Available: http://code.google.com/p/aforge

[26] G. Bradski, A. Kaehler, and V. Pisarevsky, "Learning-based computer vision with OpenCV," *Intel Technology Journal*, vol. 9, 2005.

[27] WowWee Rovio. [Online]. Available: http://www.wowwee.com/en/products/tech/telepresence/rovio

[28] K. Pitti, B. Curto, J. Garcia, and V. Moreno, "NXT workshops: Constructionist learning experiences in rural areas," in *Proc. Int. Workshop "Teaching robotics, teaching with robotics"*, Darmstadt, Germany, November 2010.

[29] E. Menegatti and M. Moro, "Educational robotics from high-school to master of science," in *Proc. Int. Workshop "Teaching robotics, teaching with robotics"*, Darmstadt, Germany, November 2010.

**Grzegorz Cielniak** is a Senior Lecturer at the School of Computer Science, University of Lincoln, UK. He obtained his Ph.D. in Computer Science from Örebro University, Sweden in 2007 and his M.Sc. in Robotics from Wrocław University of Technology, Poland in 2000. His research interests include mobile robotics, machine perception, estimation and tracking techniques, monitoring of humans and other species. Grzegorz is currently involved in teaching a number of undergraduate computer science subjects at Lincoln, including the module Computer Vision and Robotics.

**Nicola Bellotto** is a Senior Lecturer in Computer Science at the University of Lincoln, where he teaches a number of undergraduate and postgraduate modules, including Computer Vision and Robotics and Operating Systems. His research interests range from mobile robotics to cognitive perception, including sensor fusion, Bayesian estimation and embedded AI. He holds a Ph.D. in Computer Science from the University of Essex and a Laurea in Electronic Engineering from the University of Padua. Before joining Lincoln, he was a post-doctoral research assistant at the University of Oxford. Nicola has also several years of professional experience working in the industry as software developer and embedded system programmer.

**Tom Duckett** is a Reader in Computer Science at the University of Lincoln, UK, where he is also Director of the Center for Vision and Robotics Research. He was formerly a docent (Associate Professor) at Örebro University, Sweden, where he was leader of the Learning Systems Laboratory within the Center for Applied Autonomous Sensor Systems. He obtained his Ph.D. in Computer Science from Manchester University in 2001, M.Sc. with distinction in Knowledge-Based Systems from Heriot-Watt University in 1995 and B.Sc. (Hons.) in Computer and Management Science from Warwick University in 1991, and also studied at Karlsruhe and Bremen Universities. He currently teaches artificial intelligence, computer vision and mobile robotics as part of the B.Sc. program in Computer Science at Lincoln.