

Automation in the Teaching of Descriptive Geometry and CAD. High-Level CAD Templates Using Script Languages

R. Moreno¹, A. M. Bazán²

¹ University of Granada, Dpto. Expresión Gráfica, E.T.S. Ingeniería de Edificación, 18071 Granada, Spain

² UPM, Dpto. Ingeniería Civil, E.T.S. Ingeniería de Caminos, Canales y Puertos, 28040 Madrid, Spain

rmc@ugr.es

Abstract. The main purpose of this work is to study improvements to the learning method of technical drawing and descriptive geometry through exercises with traditional techniques that are usually solved manually by applying automated processes assisted by high-level CAD templates (HLCts). Given that an exercise with traditional procedures can be solved, detailed step by step in technical drawing and descriptive geometry manuals, CAD applications allow us to do the same and generalize it later, incorporating references. Traditional teachings have become obsolete and current curricula have been relegated. However, they can be applied in certain automation processes. The use of geometric references (using variables in script languages) and their incorporation into HLCts allows the automation of drawing processes. Instead of repeatedly creating similar exercises or modifying data in the same exercises, users should be able to use HLCts to generate future modifications of these exercises. This paper introduces the automation process when generating exercises based on CAD script files, aided by parametric geometry calculation tools. The proposed method allows us to design new exercises without user intervention. The integration of CAD, mathematics, and descriptive geometry facilitates their joint learning. Automation in the generation of exercises not only saves time but also increases the quality of the statements and reduces the possibility of human error.

1. Introduction

By definition, descriptive geometry is a method of studying 3D geometry through 2D images. It provides insight into the structure and metric properties of spatial objects, processes, and principles. Descriptive geometry courses cover not only projection theory but also modelling techniques for curves, surfaces, and solids, thus offering insight into a broad variety of geometric shapes [1]. ‘Learning by doing’ is an important methodological principle in this subject, and one traditional goal is to develop and refine the students’ problem-solving skills. As the drawing tools have drastically changed in the last years, this has had consequences for descriptive geometry education. CAD packages replace manual drawings. This has made the subject more interesting and attractive for pupils and students because they can now produce high-quality rendered graphics results. Of course, this development takes place at the cost of training in geometric reasoning.



The increasing importance of information technologies in the everyday world and in education makes the question of teaching descriptive geometry with the use of computer software an urgent one [2].

Computer Aided Design (CAD) is the use of computer systems to help in the creation, optimization, modification, or analysis of a design. CAD software is used to increase designer productivity, improve design quality and communications through documentation, and create databases for manufacturing. As the CAD modelling techniques become more and more advanced, it is necessary to complete product modelling and design changes faster than ever [4]. Updating assemblies that have hundreds of sub-assemblies and parts manually in 3D modelling software is very complicated and time consuming.

Undoubtedly, once a task is fully defined, computers and machines are unparalleled in executing it repeatedly with great speed and sustained accuracy. To this end, Hopgood [3] states that “computers have therefore been able to remove the tedium from many tasks that were previously performed manually”. The process referred to is also called Design Automation (DA) by various researchers. The key phrase here is that many manual tasks have been removed through DA and a natural question would be: why not remove the tedium from all manual tasks? [4].

One of the great benefits of using CAD to create our technical drawings is the ability to adapt to suit our company's processes. If you establish a technical drawing process that we perform frequently, it can be automated. If we've ever had to do the same thing with CAD twice, think about how we could automate it so we never have to do it again.

One of the easiest ways to automate a CAD process is to write a script [5]. In computer programming terms, a script is a program that will run with no interaction from the user. In AutoCAD [6], a script file is an ASCII text file that contains a set of command line instructions to follow, just like an actor reading from a script. AutoCAD script files always have the file extension '.scr'.

AutoLISP [7] is the original and most popular programming language for AutoCAD. The reason for its popularity is that it is a natural extension of the program. No additional software needs to be run, and AutoLISP can run commands that Autodesk and other developers offer in the command window.

The LISP code can be entered directly into the command window or loaded using '.lsp' or '.scr' files. Once a LISP program is loaded, the built-in functions can be executed from the command window. These functions can be executed similarly to CAD commands, but it is the programmer who decides which messages to display. It is possible to use LISP code with a command macro that is activated from the CAD user interface or from a tool on a palette.

Visual languages can be very useful for helping architecture students understand general programming concepts, but scripting languages are fundamental for implementing generative design systems [8].

It is possible to learn to draw with AutoCAD and to program with AutoLISP for AutoCAD using the manuals and online aids offered by both Autodesk (knowledge.autodesk.com) and other independent developer websites (lee-mac.com, afralisp.net, or cadtutor.com). Self-learning through tutorials and videos is very widespread and numerous websites are available to solve any questions we may raise using advanced search engines if we search for the terms 'AutoCAD' or 'AutoLisp' as appropriate.

In the design of complex engineering products, it is essential to handle cross-couplings and synergies between subsystems. An emerging technique that has the potential to considerably improve the design process is multidisciplinary design optimization (MDO) [10].

MDO requires a concurrent and parametric design framework. Powerful tools in the quest for such frameworks are DA and knowledge-based engineering. The knowledge required is captured and stored as rules and facts to finally be triggered upon request. A crucial challenge is what type of knowledge should be stored in order to realize generic DA frameworks and how it should be stored [9]. The required knowledge is captured and stored as rules and facts that will finally be activated upon demand. The aim is to shift from manual modelling of disposable geometries to CAD automation by introducing high-level generic geometry templates. Instead of repeatedly modelling similar instances of objects, engineers should be able to create more general models that can represent entire classes of objects.

With regard to Asperl [11], in CAD learning, students achieve a level of skills and knowledge appropriate for their motivation with two basic objectives: to achieve good marks in examinations and to be well prepared for further tasks. As a consequence, the position and role of teachers have to change too. Teaching in front of the audience and explaining constructions step by step should be only a small part of education in CAD and other subjects. The teacher should not be the centre of activities; it is more efficient to put the individual student in the centre. In comparison with team sports, a good teacher does not have to be the MVP CAD player but has to be the best coach.

According to Bokan [12], “rapid development of CAD/CAM software has made classical methods of Descriptive Geometry entirely obsolete. However, this discipline is still very important for strengthening one’s spatial intuition”. The introduction of modern CAD software packages in descriptive geometry improves the quality of studies; the students become more involved and interested. They also gain some practical skills that are useful in the job market. It is encouraging to learn that students can produce high quality and very useful teaching accessories.

The idea [12] is not to teach students particular methods of descriptive geometry but to solve the same spatial problems using 3D features of AutoCAD. In such a way, the geometrical solution of a problem stays the same, but the technique is no longer classical. The solution is immediately a 3D object that can be easily projected in many ways, each of which would require a separate drawing in classical descriptive geometry. This approach is also useful for strengthening a student’s spatial intuition. It is a kind of geometrical modelling but is applied to classical problems of descriptive geometry.

According to Dosen [13], “generally, online students require additional support as they need to adjust their approaches to learning. However, teaching CAD online is more challenging than teaching other subjects. On-campus students attend face-to-face tutorials and interact with their tutor who is able to interactively and visually demonstrate aspects of the CAD graphical user interface, while distance learning students rely on the communication with the lecturer as well as the available teaching materials”.

Oraifige [14] states that “the overall results support the argument that such online systems for students’ teaching, learning and evaluation can be reliably implemented; however careful planning and analysis are necessary to gain the potential benefits”.

In an effort to address the above challenges, this paper proposes the creation of high-level CAD templates (HLCts) for the manipulation of geometry and high-level analysis (HLAts) templates for concept evaluations.

2. Descriptive geometry and CAD templates

AutoCAD and other compatible applications automatically create object identifiers, hidden during the drawing process, that an average user usually does not know. These are necessary for the internal manipulation of the objects, but because of their extension and their difficult assimilation they are difficult to incorporate in procedures of descriptive geometry.

Through LISP variables it is possible to create geometric and object references similar to those traditionally used in descriptive geometry. These references will be very useful in the detailed description of graphic procedures, although with limitations: a) they do not distinguish between capital "A" and "a" lowercase; b) they do not allow "a", but "a _", "a\$", "a%", "a#", "a*", or "a&" is possible.

```
;parametric data
;
;floor: diagonal rectangular boundaries
(setq SMN '(0.0 0.0) SMX '(440.0 150.0))
;pyramid: 2 points A and B of the base, height H1 and number of sides N1
(setq A '(79.4095 29.4095 0.0) B '(79.4095 79.4095 0.0) H1 100 N1 5)
;prism: 2 points C and D of the base, height H2 and number of sides N2
(setq C '(165.981 19.4095 0.0) D '(165.981 109.4095 0.0) H2 80 N2 3)
;cone: center O3, radius R3 and height H3
(setq O3 '(215.0 60.0 0.0) R3 35.0 H3 90.0)
;cylinder: center O4, radius R4 and height H4
(setq O4 '(300.0 80.0 0.0) R4 40.0 H4 60.0)
;distant light: line R and vector VR
(setq R (list '(340 80 60) '(400 80 0)) VR (mapcar '- (cadr R) (car R)))
```

Figure 1. File P5a.scr – parametric data

In the command window, in macros, and in script files, the use of CAD drawing commands can be combined with LISP commands, functions, and variables (used as geometric and object references). They are equivalent:

`_polygon !N1 _e !A !B` ; in script files and in the command window

`(command "_polygon" N1 "_e" A B)` ; in LISP files and in script files

Notes: The comment lines are preceded by semicolons and serve to facilitate the understanding of the code. All AutoCAD commands preceded by the underscore will be executed even if the program is installed in another language. The references used in the script files or in the command window are preceded by the exclamation mark. The keyword "_edge" or its abbreviation "_e" specifies the quotation marks not in the commands but in the command function. LISP functions (in parentheses) can be used as arguments in AutoCAD commands. The commands can be used in AutoLISP using the command function.

```

;
offStatus
_erase _all
-layer _s a_data
(setq #msg1 (m-text (list 0 -6) msg1))
;
;planesurf
_surfu 0
_surfv 0
_planesurf !SMN !SMX
(setq #floor (entlast))
;
;pyramid
_polygon !N1 _c !A !B
(setq #AB* (entlast))
_pyramid _s !N1 _c !A !B !H1
(setq #pyramid (entlast))
;

;prism (=polygon&extrusion)
_polygon !N2 _c !C !D
_extrude _last !H2
(setq #prism (entlast))
;
;cone
_cone !O3 !R3 !H3
(setq #cone (entlast))
;
;cylinder
_cylinder !O4 !R4 !H4
(setq #cylinder (entlast))
;
;distant light
_line (car R) (cadr R)
(setq #R (entlast))
;
onStatus

```

Figure 2. File P5a.scr – drawing with references of parametric data

Solving an exercise in descriptive geometry using 3D modelling with CAD applications does not require advanced knowledge and, with effort, it is possible to undertake it in a self-taught way. But certain exercises require additional knowledge of descriptive geometry. Learning an exercise typology also requires the study of a topic and all possible exercises. To solve them efficiently requires the study and application of advanced procedures. By creating a set of descriptive-geometry and CAD templates that incorporate these advanced procedures and their application to any type of exercise, it can be *solved and learned to solve* the most frequent exercises of geometry.

The procedure for the creation of a generic template of descriptive geometry and CAD is as follow: We select examples from statements of current exercises. We solve first as paper-and-pencil sketches and later with AutoCAD. We extract the command history and summarize it in script files. To avoid unexpected errors when executing script commands with scripts, it is essential to disable certain drawing aids that are visible in the status line and to activate them when finished. We create reference variables with the parametric data and draw the objects. We calculate the derived references that can be reused and redraw. A later analysis will allow us to process the information in a global way using variables, functions, and commands created with LISP. Distinguishing between direct and computer processes can be of great interest. A detailed analysis and design allows us to anticipate existing relationships between surfaces through shadows or intersections. We create the necessary analysis functions. We have defined a class of exercises from an existing exercise. Finally, we carry out debugging to remove possible errors.

We present and discuss the results of two practical examples of templates: *Shadowed Developable Surfaces* and *Intersection of Planes*. The first is a shadows exercise of distant light on four surfaces: a pyramid, a prism, a cone, and a cylinder. The surfaces are supported on a horizontal plane. The shadows are solved by the separatrix contours, their projection on the horizontal plane, and their projection on the surfaces. The second exercise is the typical intersection of planes in multiview orthographic projection by means of auxiliary projections of one plane on another.

3. Results and discussion

The first template example “Practice 5 – Shadowed Developable Surfaces” consists of several types of files: a) an empty drawing template (A3.dwt); b) six scripts (P5.scr, P5a.scr, P5b.scr, P5c.scr, P5d.scr, and P5e.scr); and c) an AutoLISP (dgfun.lsp). The 3D modelling procedure is divided into several script files, organized by layers (episodes): a) P5a.scr – a_data (understanding); b) P5b.scr – b_auxiliary (analysis and design); c) P5c.scr – c_process (planning); d) P5d.scr – d_process (scan); and e) P5e.scr – e_results (verification).

```

;pyramid
(setq O1 (centroid (listpol #AB*))) V1 (elev O1 H1) V1S (shadowH V1 VR))
(setq AB* (Polygon_Lst A B N1))
(setq AB_CROWN (tan_crown_lst AB* VR))
;prism
(setq C2 (elev C H2) D2 (elev D H2) C2S (shadowH C2 VR) D2S (shadowH D2 VR))
(setq CD* (Polygon_Lst C D N2))
(setq CD_CROWN (par_crown_lst CD* VR) CD_CROWN_S (movePtList CD_CROWN C C2S))
(setq CD_CROWN2 (movePtList CD_CROWN C C2))
;cone
(setq V3 (elev O3 H3) V3S (shadowH V3 VR) O3M (midpoint O3 V3S) R3M (distance V3S O3M))
(setq IP (inters2circles O3 R3 O3M R3M) IP1 (car IP) IP2 (cadr IP))
;cylinder
(setq PIn (list (car O4) (- (cadr O4) R4) 0.0) PM (list (+ (car O4) R4) (cadr O4) 0.0))
(setq PF (list (car O4) (+ (cadr O4) R4) 0.0))
(setq P4I (elev PIn H4) P4M (elev PM H4) P4F (elev PF H4) P4IS (shadowH P4I VR))
(setq P4MS (shadowH P4M VR) P4FS (shadowH P4F VR))

```

Figure 3. File P5a.scr – analysis and design with references of parametric data

The second template example, "Practice 2 – Intersection of Planes", is composed of: a) an empty drawing template (A3.dwt); b) two script files (P2.scr and P2a.scr); and c) one AutoLISP file (dgfun.lsp).

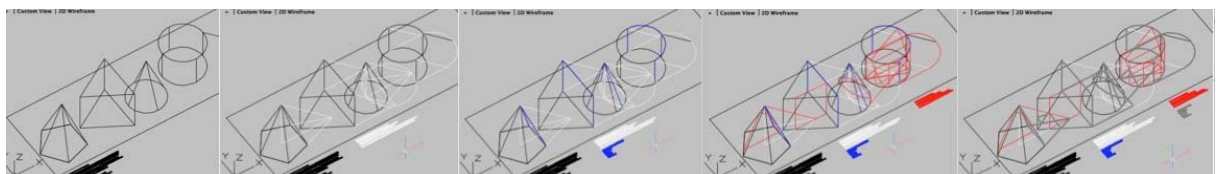


Figure 4. Screen views of episodes with script files of P5 template

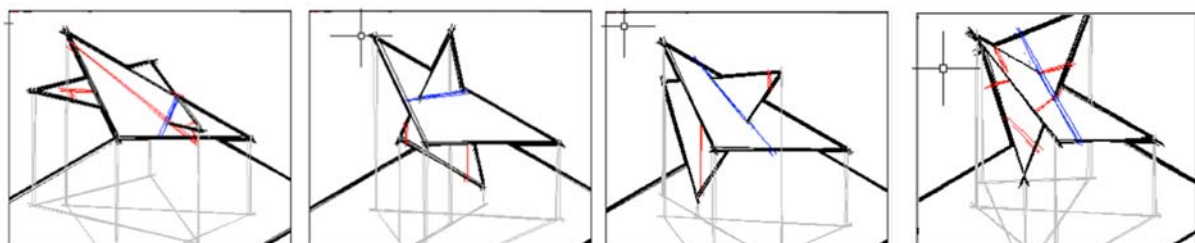


Figure 5. Screen views of P2 template

Modifying parametric data from both templates allows us to obtain different results. We plan to provide the ability to change the coordinates of reference points, number of edges of polygons, cone or cylinder radius, and height of any surface or distant light vector.

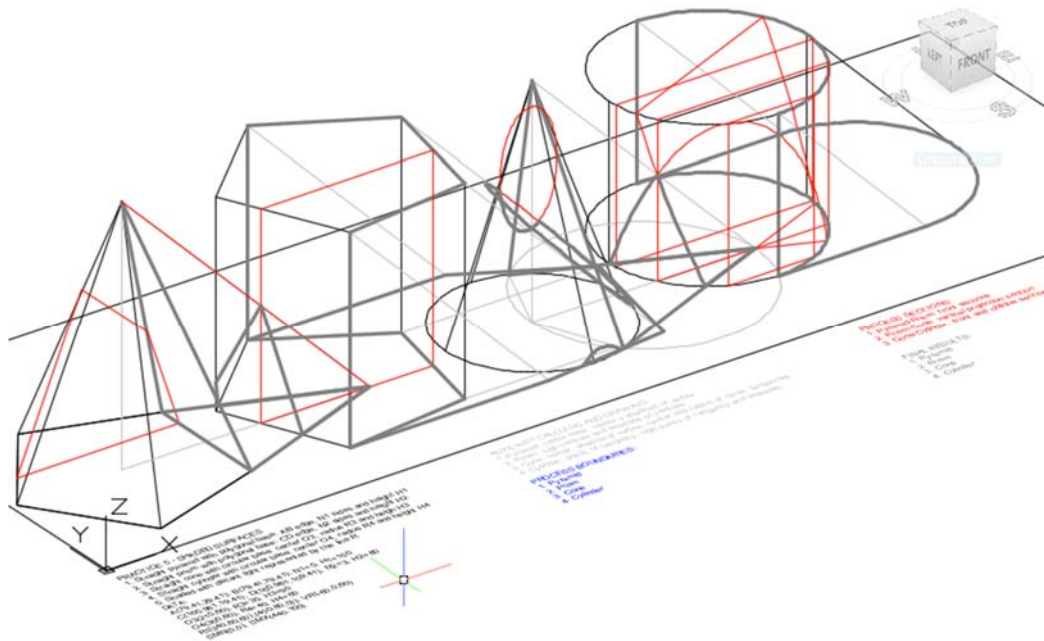


Figure 6. Results of P5 template modifying parametric data

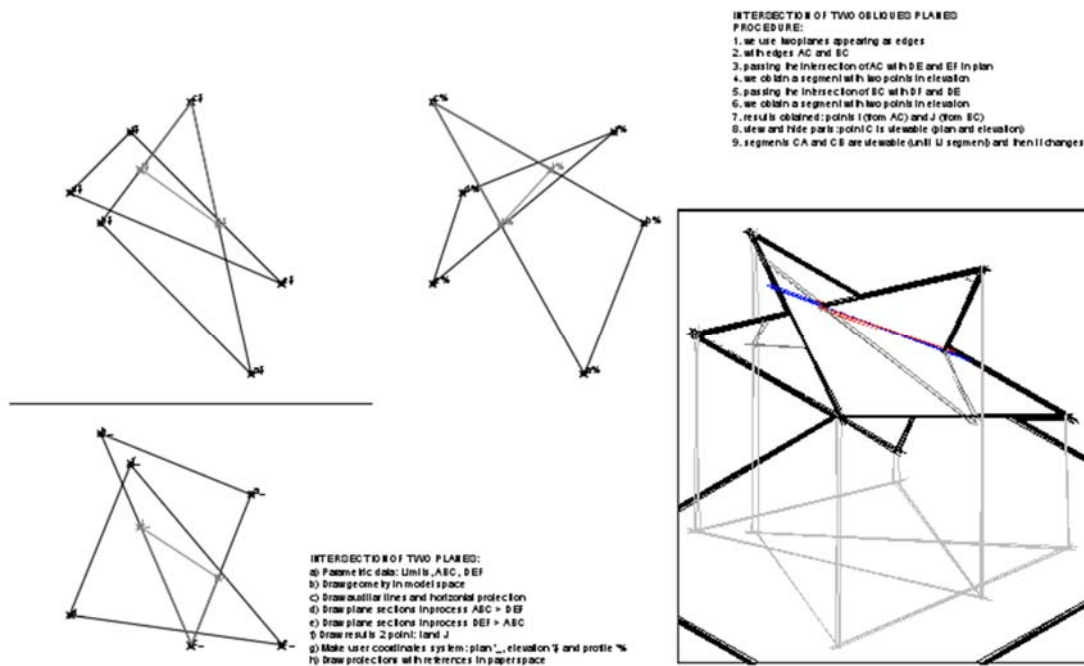


Figure 7. Results of P2 template modifying parametric data

The descriptive geometry and CAD templates, based on ontologies, represent a great advance for the automatic generation of educational resources. These templates can be used as custom practice monitors. The student is simultaneously introduced to descriptive geometry and to the complex world of CAD with its derivatives. An exercise can be repeated with multiple parametric data until it is strengthened. To avoid the exchange of results between students it is possible to change the starting data and automatically have multiple results. Although exist development platforms more advanced than AutoLISP and script files exist, they don't work in applications compatible with AutoCAD or in other operating systems (macOS and Linux). They also have the drawback that they require programming skills. Other current advanced CAD programs have powerful development environments with scripting languages, similar to AutoLISP, that enable their use as a platform for the generation of geometry and CAD templates.

Descriptive geometry and CAD templates can be useful for:

- explaining theoretical concepts using practical examples,
- allowing detailed reading of the text of the script files where the procedure followed is explained by comments,
- allowing detailed reading of the script files where the mathematical functions created are explained,
- solving the exercises provided by the teacher in a flexible way,
- testing the level of theoretical knowledge of the student through self-assessment,
- serving as a test platform for students who wish to practice successfully before being tested,
- serving as a development platform for the creation of new templates.

Only the synergy of three complementary core subjects (mathematics, descriptive geometry, and CAD) will enable students to access this knowledge in less time and with less effort. In this sense, Stachel [1] writes that "only people with a deep knowledge of Descriptive Geometry will be able to make extensive use of CAD programs" and that "the importance of mathematics continues to increase even though computers take charge of the computational work".

It can be done more in the field of multidisciplinary design optimization and incorporate programming as the fourth core subject involved in the management of descriptive geometry and CAD templates.

4. Conclusions

This document proposes a method of automation of existing procedures to generate exercises in descriptive geometry and CAD. The proposed method is analysed with several parametric data in the two examples presented, and the following conclusions can be drawn:

- 1) By using automation processes, it is possible to design exercises without user intervention.
- 2) Automation in the generation of exercises not only saves time but also increases the quality of the postulates and reduces the possibility of human errors.
- 3) Multidisciplinary optimization of design in exercises reduces learning efforts and speeds up the acquisition of graphic skills.

Expert authors make the first descriptive geometry and CAD templates, but their use and modification does not require advanced knowledge.

Acknowledgment(s)

This paper has been developed with the financing of FEDER funds under the projects TIN2013-40658-P and TIN2016-75850-R.

References

- [1] Stachel, Hellmuth. "The status of today's Descriptive Geometry related education (CAD/CG/DG) in Europe." *Journal of Graphic Science of Japan* 41.Supplement 1, pp. 15–20, 2007.

- [2] Nassery, Farid. "Autocad assisted teaching of descriptive geometry and engineering graphics." *Journal Biuletyn of Polish Society for Geometry and Engineering Graphics* 25, 2013.
- [3] Hopgood, Adrian A. *Intelligent Systems for Engineers and Scientists*, CRC Press, 2012.
- [4] Siddesh, S., and B. S. Suresh. "Automation of Generating CAD Models." *Journal of Mechanical Engineering and Automation* 5.3B, pp. 55–58, 2015.
- [5] Ambrosius, Lee. *AutoCAD Platform Customization: User Interface, AutoLISP, VBA, and Beyond*, John Wiley and Sons, 2015.
- [6] AutoCAD [computer program]. Autodesk, 2017.
- [7] AutoLISP (Part of AutoCAD) [computer program]. Autodesk, 2017.
- [8] Celani, Gabriela, and Carlos Eduardo Verzola Vaz. "CAD scripting and visual programming languages for implementing computational design concepts: A comparison from a pedagogical point of view." *International Journal of Architectural Computing* 10.1, pp. 121–137, 2012.
- [9] Tarkian, Mehdi. *Design Automation for Multidisciplinary Optimization: A High Level CAD Template Approach*, 2012.
- [10] Amadori, Kristian, et al. "Flexible and robust CAD models for design automation." *Advanced Engineering Informatics* 26.2, pp. 180–195, 2012.
- [11] Asperl, Andreas. "How to teach CAD." *Computer-Aided Design and Applications* 2.1–4, pp. 459–468, 2005.
- [12] Bokan, Neda, Marko Ljucovic, and Srdjan Vukmirovic. "Computer-aided teaching of descriptive geometry." *Journal for Geometry and Graphics* 13.2, pp. 221–229, 2009.
- [13] Dosen, Annemarie S., et al. "Teaching CAD: The challenges of online delivery to distance learning students." *Proceedings of the 37th AUBEA International Conference*, 2012.
- [14] Oraifige, Amal, et al. "An online intelligent system for teaching design technologies to engineering students." *Engineering Education* 6.1, pp. 40–51, 2011.