Software for laser projection of CAD files for the clothing industry

S Ilchev, R Andreev, Z Ilcheva and E Otsetova-Dudin

1 IICT-BAS, Acad. G. Bonchev str., bl. 2, 1113 Sofia, Bulgaria
2 University of Telecommunications and Posts, Acad. Stefan Mladenov str. 1, 1700 Sofia, Bulgaria

e-mail: svetozar@ilchev.net

Abstract. The software for laser projection of CAD files is a piece of a larger solution for laser projection of CAD data at the workbench of human operators. The software was originally intended for the clothing industry but it is suitable for other industries working with vector-based data, as well. Among our goals during the software development was the support of CAD data from various design systems used in the manufacturing industry and the design of tools for data modification to make the data suitable for laser projection. The projection is done by a laser projector that contains one or more semiconductor laser diodes and at least two rapidly rotating mirrors called scanners, which deflect the laser beam and project it on a two-dimensional surface. This principle of operation requires the modification of the CAD data, which is usually intended for use on a CNC router, plotter or another similar machine. Our software was developed to enable this modification through suitable software operations and to export the CAD data to a format supported by laser projectors. The experimental results show that we achieve our goals and that the generation of high-quality laser projections is enabled by our software.

1. Introduction

The progress in the development of semiconductor laser sources (laser diodes) has made innovative applications of laser light in the industry both technically and economically feasible. Stationary laser-generated lines or circles may be used as guiding lines for machines and instruments or as replacements for floor markings that are traditionally painted or drawn by special markers. One advantage of laser light is that no repainting or redrawing at regular intervals is needed. Another advantage is that all parts of the laser lines are clearly visible when they are projected on/near objects that are moved, processed or placed by machinery, which makes the alignment easier in comparison to traditional paint/markings, which are usually situated under the objects so they are not visible to human operators.

An enhancement of stationary laser projections are dynamic laser projections (laser animations) generated by means of a pair of small moving (rotating) mirrors called scanners, which enable the projection of arbitrary forms. This is very useful for projecting assembly instructions or other guidelines of arbitrary shape directly on the work surface used by human operators. Projections can be changed on the fly depending on the actual piece that is positioned on the work surface. Fig. 1 shows two projections of the outline of a piece of clothing generated from CAD files optimized by our software. The projections are intended to serve as reference guidelines during the processing and the inspection of the actual pieces of clothing positioned on the work surface. One issue with using laser animations is that laser projectors have particular requirements on their input data. One requirement is that the data must be present in
vector form. In addition, the maximum number of points should be limited in order to reduce the amount of flickering that human users observe during the projection. The projection of sharp corners is also problematic and has to be optimized carefully. As a result, we created our own optimization software, which deals with these issues and prepares the CAD data for laser projection.

In this paper, we address the adaptation of generic vector-based CAD data to laser projectors and we describe in detail the software that we developed with the aim of creating laser projections of data from the clothing industry. First, we present some related work. Then, we describe the conceptual design of the software and its implementation. After that, we conduct some experiments with real-world CAD data and assess the quality of the projection. We conclude the paper with a short summary of our work and give some directions for future improvements.

![Figure 1. Laser projection guidelines used during the processing/inspection of a piece of clothing.](image)

2. Related work
We present briefly some of the research related to the use of laser systems in industrial applications. The authors of [1] present the hardware aspects of a laser projection system intended for the creation of an image on a distant screen. Beam shaping and expansion that improve the laser beam quality are discussed in details. An older concept of the use of laser systems in the manufacturing industry is presented in [2]. The authors propose a new design system that makes use of lasers controlled by microcomputers. In [3], a quality control system and a strategy that uses lasers to assess if the already built prototypes are identical to the computer-generated design data is proposed. The system aims at aligning the design data to correspond to the existing prototypes. The authors of [4] propose a device for projecting images that may involve laser light sources and has a feedback mechanism that includes an optical assembly containing an optical fiber and a sensor.

An overview of laser diodes and some of their contemporary applications is presented in [5]. Some important properties of laser diodes are discussed and specific usage examples are given from the automotive, medical and multimedia fields. The improvement of the reliability of laser-based light sources and their integration in IoT (Internet of Things) applications is the topic of [6]. The authors focus on industrial applications and present the potential and the added value of cloud services and big data in relation to lasers. The use of laser sources to transmit energy and provide illumination is discussed in [7]. Laser diodes have the potential of replacing traditional light-emitting diodes due to their higher luminance and better efficiency. In [8], the use of contemporary augmented reality technologies in the design and manufacturing of products is discussed. The focus is on the need for rapid design and lean and flexible production. Flexible manufacturing and reconfigurability with a focus on the aerospace industry are the topics of [9]. A framework of technologies is discussed that includes multi-product assembly stations, cloud-based data storage and digital instructions projected by a laser system. In [10], an augmented reality system for visualizing assembly instructions directly where they are needed is proposed. This is one of the promising use cases for laser projections in the industry. The
application of augmented reality in the manufacturing industry is also discussed in [11]. The authors claim a potential for time reduction and optimization of industrial shop floors in smart factories.

Looking outside industrial environments, we may also find other application scenarios of the laser projection of computer-generated data – interactive computer-aided education [12, 13], innovative information screens for transportation networks [14], display of sensor data from both indoor and outdoor wireless sensor networks [15, 16, 17], medical manipulations assisted by robots [18], etc.

3. Conceptual design
In this section, we present the conceptual design of our software. First, we did a requirements analysis which involved visiting potential customers from the clothing industry. As a result, we identified several customer-related requirements.

The first requirement we defined is related to the input file formats that our software supports. Most customer software can export its CAD data into the standard HPGL and DXF data formats. After we looked into both data formats, we saw that HPGL has some similarities to the way laser animations are defined, saved and transmitted – usually by means of the so called ILD data format that laser projectors may use as input for the laser projections. DXF data is a bit more complex but it may still be mapped to ILD data with relative ease. Therefore, we decided as a first step to support and test the conversion of HPGL CAD data into laser projections with the plan to support DXF input files in the near future.

A second requirement that was quickly identified, is the need for additional processing and simplification of the input CAD data. Very often, the data contains some unnecessary details such as text, borders, explanatory labels, which should be removed – either by the original CAD software or by our software. One of our goals during the introduction of laser projections in the manufacturing processes was to disrupt the original flow as little as possible, so we decided to do the removal of the redundant details from the CAD data in our software. This means that the software should have a graphical user interface (GUI), which allows the manipulation and deletion of CAD data by a human user. The removal of redundant details has one additional benefit – it reduces the amount of flickering and increases the perceived animation quality. After an analysis of typical CAD data files, we determined that the raw input data cannot be used by the laser projector directly. Before the projection, it must be optimized by changing the order of the projected lines – either manually or automatically. In particular, sharp corners must be subjected to special processing, which breaks them in two halves and projects each half separately. As a result, a third requirement for our software is that it provides tools for changing the order of projection of lines and line segments, tools for breaking line segments into simpler segments and tools for merging line segments. Again, this makes necessary the presence of a GUI. Additional operations such as rotation, translation and scaling may facilitate the data projection on a predefined surface and the software should provide the corresponding tools for such operations. A fourth requirement for our software that we identified is related to the laser projector and the necessary electronics for its control. Typical laser projectors that are capable of performing laser animations are traditionally proprietary systems that may or may not conform to several standards defined by the International Laser Display Association. One of these standards is the ILDA Image Data Transfer Format (ILD data format), which allows the storage and exchange of animations for laser projectors. We decided to support the ILD data format in our software in order to be able to export and run the projections of CAD data on laser projectors, whose firmware and control software support this standard.

As part of our future development efforts, we envision that our software has a direct connection to the laser projector and controls the projections without any intermediary file exports. For this purpose, the software must be capable of running on various hardware platforms, e.g. x86/x64, ARMv8-A, possibly ARMv7E-M and other architectures. One possible use case is the integration of a single-board computer into the laser projector enclosure, which is equipped with a touchscreen and enables human operators to monitor and control the projections without the need for a separate personal computer attached to the laser projector. As a result, we want our software to support open-source Linux-based operating systems and – ideally – ARM-based microcontrollers with or without real-time operating systems. This leads us to our fifth requirement for multiplatform support. Our potential customers also
helped us identify several use cases – the most prevalent of them is as follows: a CAD file is loaded into the laser projector, it is projected for some time and then it is manually replaced with a new CAD file. Our software has first and foremost to prepare the CAD file for laser projection and in the future it will handle the change of the animations and the overall control of the laser projector.

After the requirements analysis, we define the overall software architecture. Fig. 2 shows the major software components along with the programming language chosen for each component. We talk about components and not classes or objects because our data components (HPGL_Data, DXF_Data and ILD_Data) are written in the C programming language. We chose this language because the microcontrollers that we use in our printed circuit boards for laser projector control [19] are programmed in this language and we wanted to be able to use the same source code for data import and export across all our firmware and software. As program code written in C can be compiled on almost any platform, our requirement for multiplatform support is met. For the GUI part of the software, we had to make a decision regarding the program language, the integrated development environment (IDE) and the widget toolkit. We deliberated writing it in C++ in combination with Qt5, GTK or another widget toolkit but instead we chose to use Lazarus and write it in FreePascal in combination with the LCL widget toolkit. This approach has the advantage of a much more rapid and robust development in comparison to C++ and it works excellently under both Windows and Linux including Armbian that is used on multiple popular single-board computers such as RaspberryPi. The disadvantage is that the program code is written in two programming languages and the data components need additional FreePascal definitions.

![Diagram of software components](image)

**Figure 2.** Components of the software.

The software has three data components and new ones can be easily added to support new data formats. These components read the data from the corresponding data format and convert it to a series of points that are passed to the Segment_Data component via the DataImport interface. Vice versa, a series of points can be exported to the corresponding data format via the DataExport interface. At the moment, import from HPGL and import/export from/to ILD are supported and import from DXF is planned in the near future. The central data storage of the software is handled by the Segment_Data component. This component groups series of sequential points into segments and keeps track of the number and relative order of all segments that make up the respective piece of clothing. Each point is associated with some parameters such as pen/beam color. The GUI component provides visual access to the segment data (Fig. 3). The user sees all segments and the points that belong to each segment.
Various operations supported by the GUI component allow the user to select segments or points, change the order of the selected points, add or remove points, change the point parameters, merge or split segments, etc. The changes in the segment data are passed to the Segment_Data component by means of the DataManipulation interface. Some global operations that affect all points such as rotation, translation or scaling are separated into the Data_Operations component. These operations are relatively independent from the GUI component and we may rewrite them in C in the near future. The operations are started via commands in the Transform menu of the GUI (Fig. 3) and they may have one or two parameters (e.g. angle of rotation) that are communicated through the StartOperation interface. The Laser_Control component is part of our future work and it will control the laser projection in real-time using the segment data. It provides the interface ChangeLaserState, which may be used by the GUI to start or stop the laser projection and change important parameters such as the scanner speed.

![Figure 3. Main program (GUI component) written in FreePascal.](image)

4. Software implementation

The implementation of the data components that are responsible for the CAD format support is done in C in the form of dynamic libraries (dll or so files under Windows or Linux, respectively). The libraries are used by the Segment_Data component written in FreePascal to load the CAD data when the user opens a new file in the main program window (Fig. 3).

The import of HPGL data is processed as shown in Fig. 4. First, the HPGL file is opened for reading and its contents are parsed in search for known commands. We support a subset of the HPGL commands as defined in [20], which are used in the data files of our customers. The most important and frequently encountered commands are PU and PD, which contain one or more coordinate pairs connected by straight lines. The machine tool or the laser beam should be active for PD and inactive for PU. When the HPGL_Data library encounters a PU command, it begins a new segment at the last coordinate pair belonging to the command. Then, if subsequent PD, PA or PR commands are found, all coordinate pairs belonging to the commands are added sequentially as points of the current segment. The command SP changes the pen/beam color for all subsequent points. The LB command contains text and it is processed letter-by-letter. Each letter is converted into a representation of points connected by lines according to a selected font and these points are added to the segment data. The DF, IN, IP and SC commands relate to the work area and the coordinate system. Any other commands found by the library are skipped. If no command has been found, this means that the file format is probably not HPGL or the file is empty, which is uncharacteristic as most real HPGL files that we encountered contain at least initialization commands. When the end of the file is reached, the data import is complete.
The main program (Fig. 3) shows a typical industrial file in HPGL after its data has been read by the HPGL_Data library and the segment data structure has been created. The segment data structure is a dynamic ordered array (TVector) of segments managed by the Segment_Data component. Each segment consists of a TVector of points and a reference to a TTreeNode element. Each point consists of two integer coordinates, a parameter value and a reference to a TTreeNode element. The references to the TTreeNode elements are optional and serve to provide quick access to the TTreeView graphical control element on the right side of the main program window (Fig. 3). The GUI component manages the main program window. It draws the CAD data on the screen and handles the mouse move, zoom in and zoom out operations. By left-clicking, the user may select individual segments or points and execute operations on them. The selection mode (segment or point selection) and the supported operations (e.g. adding or deleting a point or a segment, editing point coordinates or parameters, splitting or merging segments, reversing the order of points in a specific segment, copying points, reordering segments, etc.) are available by right-clicking or by pressing a graphical button from the toolbar on the left side of the program window. Fig. 3 shows a selected segment (Segment 4), whose 25 points are highlighted by small blue circles. The corresponding lines that belong to the segment are highlighted in red color. The main program window is intended to be both a viewer and an editor, which is used by human operators to delete any redundant points and to split sharp corners into two segments, which are not to be projected one after the other without blanking (switching off) of the laser beam. The CAD data in Fig. 3 contains a total of 201 points, which is a low enough number, but the rectangular element on the right side is redundant and should be removed. There are several sharp corners that should be inspected and split if necessary. After that, the segments can be saved as an ILD data file by the ILD_Data component. Optionally, the Transform menu contains Rotate Left/Right operations handled by the Data_Operations component, which rotate the whole element and transform the coordinates of all points accordingly. Translation and scaling operations will be added in the near future. In the Transform menu, there is also a command for the automatic optimization of the segment data, which deletes duplicate data points or segments and merges sequential segments. Our plan is to automate the optimization of sharp corners in the future leaving only the removal of redundant points and the final inspection to human operators.

5. Experimental results

Our experimental results show that the optimized CAD data is suitable for a high-quality laser projection provided that the hardware of the laser projector and specifically the scanners have the necessary positioning accuracy. Fig. 1 shows two examples of laser projections that were requested by customers in the clothing industry. For these first experiments, we used a single-color laser projector which included the popular DT-40 scanners, whose downside is the presence of some temperature drift and
some non-linearity that depends on the projection angle. The laser projector was controlled by standard software, which received as input our optimized ILD files. The projections in Fig. 1 exhibit some slight inaccuracies that are typical for the construction of the DT-40 scanners but the overall projection quality is pretty good and acceptable for many industrial applications. If the temperature drift and the non-linearity are compensated, e.g. through recalibration at regular intervals, the projection accuracy is improved. Furthermore, after reaching temperature equilibrium, the scanners stabilize but recalibration is often inconvenient and slow and our plan is to replace them in the near future with scanners that are specifically developed for industrial use and long-term operation and have better linearity.

We tested our software under both Windows and Linux. It performed as intended and we are very happy with the ease and speed of the CAD data optimization process. CAD data was imported successfully from several different design systems used by customers. To obtain the projections in Fig. 1, we had to delete some points and change the order of projection of some of the sharp corners. Then, we exported the modified data to an ILD file and passed it to the laser projector. As a preliminary step for our future work, we were also successful in loading HPGL files directly in our prototype smart compact laser system [19] using the developed C libraries.

6. Conclusions
The paper discusses the design, implementation and testing of our software for laser projection of CAD files for the clothing industry. First, we identify some of the requirements for handling and modifying CAD data. Then, we describe the conceptual design of the software and its architecture. After that, we discuss briefly its implementation as a set of software libraries and a multiplatform program with a graphical user interface. Finally, we present some experimental results and conclude the paper with some directions we identified for our future work. First, we plan the support of DXF CAD data to improve the compatibility of our software with other design and production systems. We also plan to extend our software to enable the direct control of laser projectors without the need for any intermediary data exports. We will also continue working on the automation of the data optimization with the goal of integrating parts of it in the firmware of the laser projector.

Acknowledgements
This research is supported by the National Scientific Program “Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)”, financed by the Ministry of Education and Science.

References
[1] Kurtz, A., Kruschwitz, B., Ramanujan, S. (2003). Laser projection display system. Patent US6577429B1, URL: https://patents.google.com/patent/US6577429B1/en.
[2] Ahonen, S. (1995). Laser projection system for truss manufacturing. Patent: US5430662A, URL: https://patents.google.com/patent/US5430662A/en.
[3] Morden, J., Rueb, K. (2005). Laser projection system, intelligent data correction system and method. Patent US7463368B2, URL: https://patents.google.com/patent/US7463368B2/en.
[4] Melville, C., Johnston, R., Lee, C., Seibel, E., Schowengerdt, B. (2014). Scanning laser projection display devices and methods for projecting one or more images onto a surface with a light-scanning optical fiber. Patent: US8757812B2, URL: https://patents.google.com/patent/US8757812B2/en.
[5] Basu, C., Meinhardt-Wollweber, M., Roth, B. (2013). Lighting with laser diodes. Advanced Optical Technologies, 2(4), 2013, pp. 313-321, ISSN: 2192-8576.
[6] Ding, J., Liu, J., Wei, X., Xu., J. (2017) The SMAT fiber laser for industrial applications. Proceedings SPIE 10085, Components and Packaging for Laser Systems III, 100850R (22 February 2017), ISSN: 0277-786X, DOI: 10.1117/12.2250459.
[7] Chatzizyrli, E., Tinne, N., Lachmayer, R., Neumann, J., Kracht, D. (2017). Modeling of photoluminescence in laser-based lighting systems. Proceedings SPIE 10603, Photonics,
Devices, and Systems VII, 1060318, 2017, ISSN: 0277-786X, DOI: 10.1117/12.2292735.

[8] Plinta, D., Krajčovič, M. (2015). Production System Designing with the Use of Digital Factory and Augmented Reality Technologies. Szewczyk R., Zielniński C., Kaliczyńska M. (eds) Progress in Automation, Robotics and Measuring Techniques. ICA 2015. Advances in Intelligent Systems and Computing, vol 350. Springer, Cham, Print ISBN 978-3-319-15795-5, Online ISBN 978-3-319-15796-2, DOI: 10.1007/978-3-319-15796-2_19.

[9] Jackson, K., Efthymiou, K., Benton, J. (2016). Digital Manufacturing and Flexible Assembly Technologies for Reconfigurable Aerospace Production Systems. Proceedings CIRP, Vol. 52, 2016, pp. 274-279, ISSN 2212-8271, DOI: 10.1016/j.procir.2016.07.054.

[10] Paclke, V. (2014). Augmented reality in the smart factory: Supporting workers in an industry 4.0 environment. Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, 2014, pp. 1-4. DOI: 10.1109/ETFA.2014.7005252.

[11] Syberfeldt, A., Danielsson, O., Gustavsson, P. (2017). Augmented Reality Smart Glasses in the Smart Factory: Product Evaluation Guidelines and Review of Available Products,” in IEEE Access, vol. 5, pp. 9118-9130, 2017, DOI: 10.1109/ACCESS.2017.2703952.

[12] Paunova-Hubenova, E., Terzieva, V., Dimitrov, S., Boneva, Y. (2018). Integration of Game-Based Teaching in Bulgarian Schools – State of Art. Proc. of 12th European Conference on Game-based Learning ECGBL 2018, Sophia Antipolis, France, pp. 516-525, ISSN 2049-0992.

[13] Savov, T., Terzieva, V., Todorova, K. (2018). Computer Vision and Internet of Things: Attention System in Educational Context. Proc. of the 19th International Conference on Computer Systems and Technologies (CompSysTech’18), Boris Rachev and Angel Smrikarov (Eds.). ACM, New York, NY, USA, Ruse, Bulgaria, pp. 171-177, DOI: 10.1145/3274005.3274014.

[14] Boneva, Y. (2018). Optimization of Car Traffic Flow on Intersections Regulated by Traffic Lights Through the Simulation Environment AIMUSUN. Academic journal Mechanics Transport Communications, ISSN 1312-3823 (print), ISSN 2367-6620 (online), Vol. 16, issue 2, 2018, Todor Kableshkov University of Transport, Bulgaria, pp. I-1–I-9.

[15] Alexandrov, A., Monov, V. (2017). Method for indoor localization optimization of AoA based mobile devices. Proc. of 12th Annual Meeting of the Bulgarian Section of SIAM BGSIAM’17, December 20-22, 2017, Sofia, Bulgaria, 2017, ISSN:1313-3357.

[16] Alexandrov, A., Monov, V. (2018). Method for Adaptive Node Clustering in AD HOC Wireless Sensor Networks. Vishnevskiy V., Kozyrev D. (eds) Distributed Computer and Communication Networks. DCCN 2018. Communications in Computer and Information Science, vol. 919, Springer, ISBN 978-3-319-99447-5, DOI: 10.1007/978-3-319-99447-5_22.

[17] Alexandrov, A., Monov, V. (2014). ZigBee smart sensor system with distributed data processing. Proc. of the 7-th IEEE Conference Intelligent Systems, Warsaw Poland, Vol. 2, pp. 259-268, September 24-28, 2014., Advances in Intelligent Systems and Computing, Springer, vol. 323, ISBN 978-3-319-11309-8, DOI: 10.1007/978-3-319-11310-4_23.

[18] Ivanova, V., Bachvarov, D., Boneva, A. (2018). An Advanced Robot System for Diagnostic and Therapeutics Tasks with Application in Laparoscopic Surgery. Journal of Computer Engineering & Information Technology, Vol. 7, Issue 2, ISSN: 2324-9307 (Online), DOI: 10.4172/2324-9307.1000198, SciTechnol, 2018, London, United Kingdom, pp. 1-9.

[19] Ilchev, S., Petkov, D., Andreev, R., Ilcheva, Z. (2019). Smart Compact Laser System for Animation Projections. Cybernetics and Information Technologies, vol. 19, no. 3, pp. 137-153, Bulgarian Academy of Sciences, Print ISSN: 1311-9702, Online ISSN: 1314-4081, DOI: 10.2478/cait-2019-0030.

[20] Hewlett-Packard Company. The HP-GL/2 and HP RTL Reference Guide: A Handbook for Program Developers, Addison-Wesley, 1996, ISBN: 0-201-63325-6.