Automatic Data Extraction to Support Management Application

Rui Melo
Research Center in Digitalization and Intelligent Robotics (CeDRI),
Instituto Politécnico de Bragança, Portugal, and
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Portugal
ruimelo@ipb.pt

Clara B. Vaz
Research Center in Digitalization and Intelligent Robotics (CeDRI),
Instituto Politécnico de Bragança, Portugal, and
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Portugal
clvaz@ipb.pt

Ana I. Pereira
Research Center in Digitalization and Intelligent Robotics (CeDRI),
Instituto Politécnico de Bragança, Portugal, and
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Portugal
apereira@ipb.pt

Abstract — **When designing a custom-made product it is important to provide the customer with a budget that resembles the final price. In this work it will be developed a simple application in *Python* to perform automatic data extraction from computer aided design (CAD) files to estimate multiple linear regression models with the intent of obtaining a more accurate cost estimate. The application will provide an estimate of the amount of raw material needed and time taken to produce a simple inflatable and related products.**

Keywords – Tridimensional mesh; CAD; Python; Multiple linear regression; Graphical user interface.

I. INTRODUCTION

For companies that design custom-made products giving an accurate budget to the customer is of utmost importance so that the customer can decide with real information. Providing an inaccurate cost estimate might damage the company's reputation and credibility, leading to reduced competitiveness and income as well as loss of customers. This research paper was developed in collaboration with a company that manufactures inflatables, that produces custom-made products with different levels of complexity. Therefore, in order to support this company, a graphical user interface that calculates an accurate cost estimate to provide the customers was developed. The user-friendly application (named iBudget) was built in Python utilizing multiple linear regression models [1] based on information extracted from CAD files from the company's database. The application was installed in the company's computer in order to support the budget task for inflatables with a low level of complexity. iBudget is easy to use, enabling to automate the process of obtaining cost estimate based on the relevant features of each custom-made inflatable. Currently, these features are retrieved from the CAD files developed during the budget procedure. In order to automate the process of retrieving the required features from CAD file, this work presents a strategy for automatic data extraction.

The present paper provides a comprehensive analysis of our research findings. Section II presents a thorough review of the relevant literature for our research. In Section III, we describe the dataset that a local manufacturer provided. Section IV presents a brief summary of the previous work, the data analysis, and its findings. In Section V, the discussion of the application's development and the methods used for automatic data extraction are presented. Finally, the last section summarizes our results and discussions and their significance in relation to the broader research context.

II. LITERARURE REVIEW

Although no works were identified using machine learning techniques to predict the cost of manufacturing inflatables, several studies have applied these techniques to other products, mostly built of metal components.

F. Ning et *al.* [2] have managed to predict the cost of production with a 6.34% mean absolute percentage error (MAPE) utilizing a tridimensional convolutional neural network. To archive this result, the CAD model was rotated in several directions in order to increase the amount of data and afterwards these models were converted into voxel volumes (the tridimensional equivalent of a pixel [3]) which were used to train the neural network. Even though the results look promising, this method required a large amount of data to archive this result. In this work it was used up 400 000 voxel volumes.

2023 18th Iberian Conference on Information Systems and Technologies (CISTI)
20 – 23 June 2023, Aveiro, Portugal
ISBN: 978-989-33-4792-8

Zhang et *al.* [4] proposed a different approach that was able to obtain nearly similar results with a MAPE of 7.53%, using a significantly less data, approximately 3072 CAD models. To archive this, the authors extracted the features from the CAD files and used them to train a graph neural network, including as well the raw material of components in the form of a one-hot vector.

The multiple linear regression has also been used in several research works to predict the budget, for example Sayadi et al. [5] used it to predict the budget of machines used in underground mining, transport, loading and unloading with a MAPE of 6.87%. To archive this result, a dataset with 16 different products was used.

C. Hai [6] explored the multiple linear regression to forecast the cost of construction projects in which it was archive a maximum deviation rate of 4.8%.

III. Methodology

The company under study works in a small industry related to the textile sector where they manufacture inflatable-based products, such as advertising products, decorative products, and agricultural tarpaulins in which cutting and sewing processes are required. These products are classified into three distinct families, with similar production processes and raw materials as described in [1]. Family set I comprises advertising-related products, such as tents, totems, and flags; family set II includes decorative inflatables, and family set III is made up of agricultural tarpaulins. The main raw material for the production of products are canvas, nets, and fabrics. These raw materials are received in rolls which can be described by width and length, since the roll width is always the same, the material spent can be measured by its own length in the unit of linear meters. Globally, these inflatable products involve cutting and sewing processes.

The data provided included 83 CAD models as well as other component costs that contributed to the final price. By analyzing the data, it was possible to conclude that the variables that most affect the production cost are the total linear meters of material spent (LM), the cutting time (CT), and the sewing time (ST) [1]. Hence, a multiple linear regression model was created to estimate each one of these production factors based on features captured manually from the CAD models.

When extracting data from the CAD model, it was considered geometric parameters that are relevant to each of the production factor (LM, ST, and CT). Following [1], the independent variables correspond to the geometric features manually extracted which include, area (A), perimeter (P), weight (W), Average height (H) and volume (V).

A. Model

A multiple linear regression model makes a prediction by simply computing a weighted sum of the input features, plus the intercept term (β_0) and the random error component (ε) [7]. As described by the formula (1), where β_i represents the regression parameters, x_i is the geometric feature and i simply represents the number of the features included.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_i x_i + \varepsilon \quad (1)$$

Thus, three models were created in order to estimate the dependent variables such as LM, ST, CT.

For evaluating the model performance, the dataset was split 80% for training data, and 20% for testing data. This is a commonly used split for dividing the data, however this proportion will depend on the size of the dataset [8].

Before training the multiple linear regression (MLR) model for each dependent variable, it is necessary to select the independent variables that are statistically significant for each model. After an extensive variable selection described in [1], it was concluded that the best MLR model for predicting the total LM, should include all the geometric features defined in Section II, named *A*, *P*, *W*, *H* and *V*. The best MLR model for predicting the cutting time should include the *A*, *P* and *H*. And the best MLR model for predicting the sewing time should include the *A*, *V*, and *H*. Afterwards, each MLR model was trained to predict the production cost of the lowest complexity level of inflatables.

This paper extends the previous research presented in [1], by implemented three MLR models into an application built in *Python* which enables to streamline the process of predicting the manufacturing costs, making them easy to obtain by a user. Additionally, since in [1] the required geometric features were extracted manually from the CAD file to feed each MLR model, this work develops a prototype to automatically retrieve them from a CAD file as described in the next section.

IV. Application

The application serves the purpose of making it as easy as possible to estimate the cost of a product, therefore it should require the minimum effort from the user, thus everything that is possible to automate must be implemented.

This section will be divided into two sections. The first one will address the prototype by describing the process used to automatically extract all the relevant geometric features proposed by [1]. And the second one will describe the graphical user interface (GUI) that was developed and how it is organized.

A. Automatic data extraction prototype

The *.step* file type was selected to be used to automate the data extraction due to being an industry standard as described by International Organization for Standardization (ISO) 10303 [9].

Initially the CAD file will be loaded and the area (*A*) and perimeter (*P*) will be extracted directly utilizing the tools provided by FreeCad's *Python* application programming interface (API). The mass will be calculated by multiplying the total area (*A*) of the inflatable by the surface density (Kg/m²) of the material (*SD*) it is built from, as described by the formula (2),

$$W = A \times SD. \quad (2)$$

The average height (*H*) can be calculated by casting multiple rays onto the mesh and calculating the intersection. To perform this calculation the CAD file is converted into a

2023 18th Iberian Conference on Information Systems and Technologies (CISTI)
20 – 23 June 2023, Aveiro, Portugal
ISBN: 978-989-33-4792-8

triangular mesh, which is a tridimensional object representation consisting of a collection of vertices and polygons [10]. To achieve this objective, the libraries *trimesh* and *pyembree* were utilized. *Trimesh* is used to perform several actions on the triangular meshes while the *Pyembree* provides *Python* bindings for *Intel embree*, a high-performance library used for performing various ray operation, such as ray-triangle intersection, described as follows.

In order to calculate the intersection, it is necessary to consider the starting point for all of the rays and the vectors that will indicate in which direction the points will move. For the vector array all the values were set pointing downwards.

For the starting points, it was used the values from the bounding box (see in Figure 1), a rectangle box that is built from the maximum and minimum values of the points on all axis of a 3D model of the inflatable. By utilizing the bounding box, it is possible to quickly determine the maximum and minimum coordinates of the model.

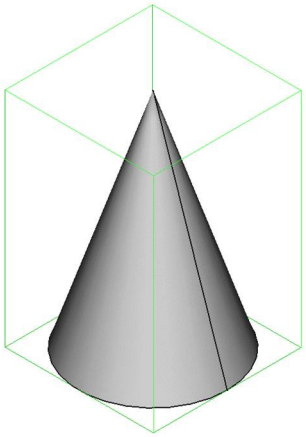

Figure 1 - Example of a bounding box

The ray starting points were all placed slightly above the mesh, that being the maximum Z axis coordinate. Then an array of points were created inside the X and Y bounds of the bounding box, as shown in Figure 2. These are the positions where the rays will be cast from onto the triangular mesh.

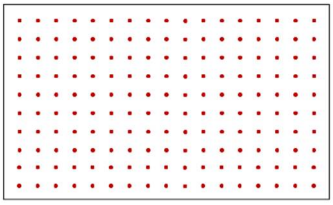

Figure 2 – Example of Array of points scattered across the XY face of the bounding box.

The distance between each point will affect the accuracy of the value extracted associated to the H. Lower distance between points will increase the accuracy, but it will increase the computing time. So, a tradeoff must be made between accuracy and time to calculate intersections.

After calculated the starting points and vectors, the rays are cast downwards and the intersections of the triangles with the rays are calculated. This returned a vector with the coordinates of all the intersections. The height was obtained by extracting the Z value and subtracting it by the minimum Z value from the bonding box. Afterwards, the average height (H) is calculated.

Due to the way the company designs inflatables, the CAD models are not watertight, so the volume information can only be obtained via approximation.

In order to calculate the approximation of the volume (V), the matrix containing the height of each point will be reutilized. Thus, by multiplying the square of the distance between points times the height, the volume of a parallelepiped will be calculated in the coordinate of the ray.

The approximation for the volume of the model is obtained using the formula (3), where D_i is the distance between the points, and H_i is the height of the intersection of a ray with the mesh.

$$V = \sum_{i=1}^{n} {D_i}^2 \times H_i. \quad (3)$$

However, this method has the disadvantage of ignoring concavities in the middle and under the 3D model, as well as under overhanging features.

If tridimensional model was reconstructed from adding up all the parallelepipeds, it would look like Figure 3.

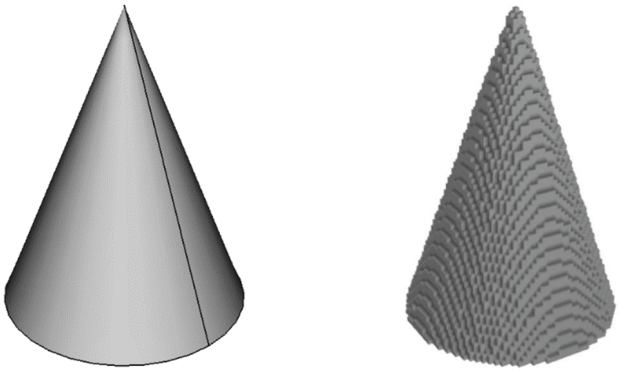

Figure 3 - Example of original CAD model compared to the model approximation created from the height on all points.

In short, the developed Python code was designed to automatically capture geometric features from a product, using the CAD files as input, and providing the area, perimeter, weight, average height, and volume as output, that later will be used in order to calculate the regression.

B. Graphical user interface

The user interface is designed in a simple and intuitive way to estimate the most influential production factors such as the LM indicated by "Meters of material", ST indicated by "Sewing time", and CT indicated by "Cutting time", and consequently, the component costs are calculated. The layout of the user interface is divided into three sections as shown in Figure 4.

ISBN: 978-989-33-4792-8

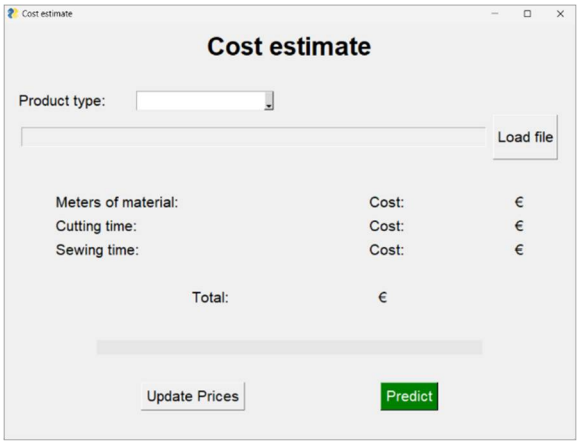

Figure 4 - Application Layout.

The top section is where the user should insert the necessary parameters. The first one is a drop-down menu to indicate what type of product it is, named "Farming", "Decorative", "Advertising", "Inflatable", and "Net". This will be used to determine what surface density to use when calculating the weight. The second one is to load the CAD file. There is a button that will open the file picker that is configurated to only show files of the extension *.step*.

The middle section is where the calculated component costs will be displayed based on the total meters of material used and the time taken to cut and to sew the inflatable, which are also visible. Since the company operates on the Eurozone, the cost will always be indicated in euro currency.

The last section has a progress bar to give feedback to the user and two buttons. One of the buttons will open a submenu to give the user the option of updating the unit price per linear meter, the unit price per hour in sewing stage and the unit price per hour in cutting stage, as shown in Figure 5. The second button performs the data extraction as well as the estimation for each component cost, displaying the values on the screen. This button only executes these tasks if there are no missing inputs, otherwise it will give a user warning.

If the submenu shown in Figure 5 is open, it will read the current saved unit prices from a *.csv* file and display them on screen, in which these values can be edited. There are also two buttons which were color coded to make the application more user friendly. If the cancel or close button is pressed the submenu is closed without saving. If the save button is pressed the values in the input boxes will be written to the *.csv* file.

Figure 5 - Application submenu.

C. Testing

According to [1], when using the test dataset, the regression models obtained the following mean absolute error (MAE), mean square error (MSE), and root mean square error (RMSE):

	MAE	MSE	RMSE
Amount of material	5.462	46.196	6.797
Time to cut	0.241	0.122	0.349
Time to sew	1.546	3.485	1.867

Table 1 - Error obtained for the 3 models from the test dataset [1] .

The application was tested for a random product, in which it was possible to obtain the following values:

Inputs	Extracted values
Area (m^2)	40.48
Perimeter (m)	85.52
Weight (kg)	31.17
Average height (m)	1
Volume (m^3)	1

Table 2 - Values inserted into the regression model.

Outputs	Real	Predicted
Linear meter	25 meters	28.05 meters
Cutting time	30 minutes	42 minutes
Sewing time	3 hours and 19 minutes	5 hours and 3 minutes

Table 3 - Results obtained from the application.

For this product the result was close to the real one, with the exception being the sewing time. Overall the application managed to predict the values reasonably, but there is room for improvement that should be addressed in future works.

V. Conclusion

The results of this study have demonstrated that the application developed can provide an easy way to predict costs based on geometrical and physical properties, automatically captured, which can support the budget step of the company that designs custom-made inflatables in order to obtain a more accurate cost product, and consequently a suitable sale price to the customer.

Since the dataset used for this study was small, it is suggested to retrain a new model once more data became available, or even implementing a script that automatically retrains the model ounce a certain number of new products are inserted into the company's database. It is also necessary to test the application in the real world in order to validate the multiple regression models and the automatic data extraction prototype. Recently, the first version of the application has been installed

ISBN: 978-989-33-4792-8

in the company's computer in order to validate the model. In the future, it will be developed a database for a more complex inflatables, by using the proposed prototype in which the data related to them will be extracted in automatic way from CAD models. Overall, the use of multiple linear regression in this context has significant potential for improving the cost estimation process in the inflatable industry, and further research could explore the potential for applying this approach to other industries and applications.

ACKNOWLEDGMENT

This work has been supported by Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020), SusTEC (LA/P/0007/2021), and NORTE-01-0247-FEDER-072598 iSafety: Intelligent system for occupational safety and well-being in the retail sector.

REFERENCES

[1] L. H. Matte and C. B. Vaz, "Decision Making System to Support the Cost of Ordered Products in the Budget Stage," pp. 486–501, 2022, doi: 10.1007/978-3-031-23236-7_34.

[2] F. Ning, Y. Shi, M. Cai, W. Xu, and X. Zhang, "Manufacturing cost estimation based on a deep-learning method," J Manuf Syst, vol. 54, pp. 186–195, Jan. 2020, doi: 10.1016/J.JMSY.2019.12.005.

[3] D. Bell, "Voxel | Radiology Reference Article | Radiopaedia.org." https://radiopaedia.org/articles/voxel (accessed Feb. 25, 2023).

[4] H. Zhang et al., "A novel method based on a convolutional graph neural network for manufacturing cost estimation," J Manuf Syst, vol. 65, pp. 837–852, Oct. 2022, doi: 10.1016/J.JMSY.2022.10.007.

[5] A. R. Sayadi, A. Lashgari, and J. J. Paraszczak, "Hard-rock LHD cost estimation using single and multiple regressions based on principal component analysis," Tunnelling and Underground Space Technology, vol. 27, no. 1, pp. 133–141, Jan. 2012, doi: 10.1016/J.TUST.2011.08.006.

[6] C. Hai, "Construction and Application of Multiple Linear Regression Model for Construction Project Cost," Proceedings - 2021 International Conference on Advanced Enterprise Information System, AEIS 2021, pp. 54–58, 2021, doi: 10.1109/AEIS53850.2021.00017.

[7] A. Géron, "Linear Regression," in Hands-On machine learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed.O'Reilly, 2019, pp. 114–116.

[8] A. Géron, "Testing and validanting," in Hands-On machine learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed.O'Reilly, 2019, pp. 31–31.

[9] International Organization for Standardization, "STEP Module and Resource Library (SMRL) v9," 2014. https://www.iso.org/obp/ui/#iso:pub:PUB100467 (accessed Feb. 26, 2023).

[10] B. Furht, "Mesh, 3D," Encyclopedia of Multimedia, pp. 406–407, Jun. 2006, doi: 10.1007/0-387-30038-4_126.

ISBN: 978-989-33-4792-8