RICA, E., ALVAREZ, S., MORENO-GARCIA, C.F. and SERRATOSA, F. 2022. Zero-error digitisation and contextualisation of piping and instrumentation diagrams using node classification and sub-graph search. In Krzyzak, A., Suen, C.Y., Torsello, A. and Nobile, N. (eds.) *Structural, syntactic, and statistical pattern recognition: proceedings of the 2022 Joint International Association for Pattern Recognition (IAPR) international workshops on statistical techniques in pattern recognition, and structural and syntactic pattern recognition (S+SSPR 2022)*, 26-27 August 2022, Montréal, Canada. Lecture notes in computer science, 13813. Cham: Springer [online], pages 274-282. Available from: https://doi.org/10.1007/978-3-031-23028-8_28

Zero-error digitisation and contextualisation of piping and instrumentation diagrams using node classification and sub-graph search.

RICA, E., ALVAREZ, S., MORENO-GARCIA, C.F. and SERRATOSA, F.

2022

Zero-Error Digitisation and Contextualisation of Piping and Instrumentation Diagrams using Node Classification and Sub-graph Search

Elena Rica[1], Susana Álvarez[1], Carlos Francisco Moreno-García[2], and Francesc Serratosa[1*]

[1]Universitat Rovira i Virgili, Tarragona, Catalonia, Spain.
[2]Robert Gordon University, Aberdeen, Scotland, United Kingdom.
[*]Corresponding author: Francesc Serratosa (francesc.serratosa@urv.cat)

Abstract

Thousands of huge printed sheets depicting engineering drawings keep record of complex industrial structures from Oil & Gas facilities. Currently, there is a trend of digitising these drawings, having as final end the regeneration of the original computer-aided design (CAD) file, which can be better visualised and analysed through diverse computer applications. Most efforts in literature and commercial applications have focused on converting these sheets into CAD files in an automated way. Nonetheless, this needs to be a zero-error process; as the final CAD will always be verified by an engineer for integrity and inspection. In this paper, we present a method that, on the one hand, highlights which components in the CAD are most likely to have been incorrectly identified, and on the other hand, facilitates the engineer to search some groups of components in these huge assets. These techniques are based on graph embedding, computer neural networks and sub-graph matching.

Keywords Piping and Instrumentation Diagram, Automatic Validation, Sub-graph Matching, Graph Embedding.

1 Introduction

Piping and Instrumentation Diagrams (P&IDs) are used to represent the structure and functionality of Oil & Gas facilities such as oil rigs and plants. P&IDs contain similar shapes to other complex engineering drawings such as circuit, architectural, mechanical, telephone manhole and chemical plant depictions. P&IDs are mostly generated by means of computer-aided design (CAD) tools and kept in an electronic record. However, in the past they were manually drawn on paper or using tools that are incompatible with modern software. Since these facilities are huge and composed of thousands of electric, electronic or mechanical components connected by a vast network of pipelines, printed handbooks composed of thousands of pages are required to depict them. Figure 1 shows a snippet of one sheet and portrays the complexity of a P&ID.

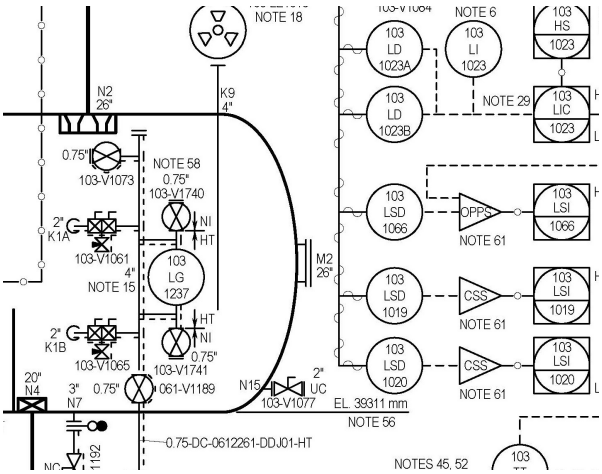

Figure 1: An example of a P&ID.

Analysing a facility using a P&ID handbook is an extremely complex process, due to the page quality and the variability of the electric, electronic and mechanic components. While several tools have been presented in recent years to generate a CAD file from these drawings in automated ways [8], the possibility of symbol miss-identification during the digitisation or that some properties have not been correctly associated to certain components becomes high [1], [3]. Thus, it is expected that this process is not perfect and therefore, most systems enable human interaction to validate the symbol identification, connection and property association. In practice, the final CAD register is always verified by an engineer due to the need of being a zero-error process. Figure 2 shows a general flow diagram of the classical approach to extract a CAD given a sheet of P&ID. The automatic module is composed of two main steps: digitisation (converting the pulp and paper drawing into a digital register or parts count) and contextualisation (understanding the interaction between the digitised shapes, such as how symbols connect to each other and the text that describes each process,

amongst others).

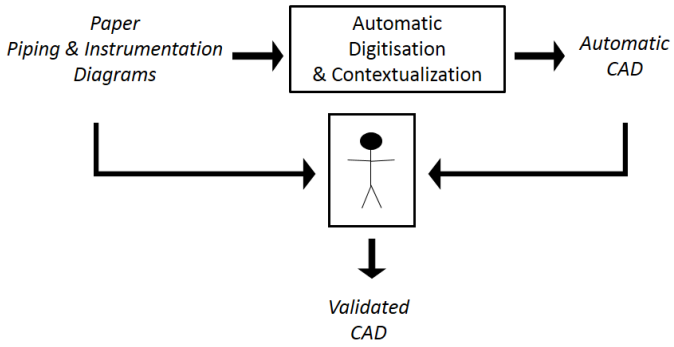

Figure 2: The process of deducing the CAD document, in which a human is involved to validate the data.

In this paper, we propose the integration of a couple of previously published tools [11, 12] for engineers and risk analysts to reduce the amount of effort needed to validate the CAD model towards creating a zero-error digitisation and contextualisation process. The goal of the first tool, depicted in Figure 3, is to aid in the validation of the automated digitisation process by ensuring that the engineer does not need to look at the whole diagram, but only at the highlighted components, which are the ones that have a chance of having incorrectly identified by the automatic method.

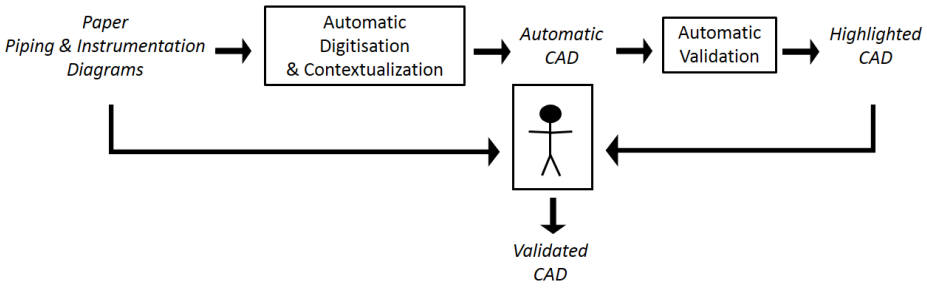

Figure 3: Our model for automatic detection of possible incorrectly identified components and final human validation.

The aim of the second tool is to aid in the contextualisation by facilitating the search for a particular configuration of connected components. Figure 4 shows a P&ID sheet analysed by our proposed application. Note that CAD applications usually have enabled functionalities to search for specific components by their identity number and visualise them in their locations [7, 9, 10, 5, 2, 14]. Nevertheless, while inspecting or analysing the gas facilities, engineers are usually interested in some structures, composed of a small set of connected components,

instead of a specific type of component. For instance, they want to detect the appearances of structures that include a *valve check* connected to two *general valves* and a *butterfly valve*. Thus, engineers want to query a structure instead of a component and visualise it in the P&ID. Note that the aim of this example query is not to return the locations of the exact appearance of the specific structure, but the locations in structures similar to the one queried may appear. Thus, considering this example, it could be interesting to return the locations of structures composed of a *valve check* connected to one or three *general valves* and one or two *butterfly valves*.

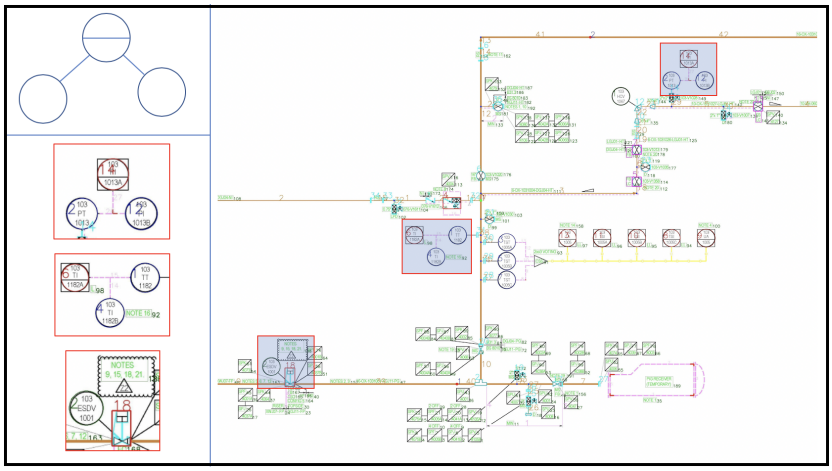

Figure 4: Screenshot of the second tool proposed. Upper left corner: query structure. Lower left corner: returned structures ordered by increasing distance. Right: a portion of the P&ID in which three locations of the query have been detected.

Both tools fit with the topological challenge of P&ID contextualisation [6], [8], whose objective is to understand the connectivity of the symbols. In comparison, we have been able to identify some commercial CAD applications[1] which work with P&IDs and that have online tools to quickly visualise the network of components. Moreover, we have presented some initial proof of concept tools that perform similar functions, such as NetVis or Netlist2CAD[2]. While these tools offer the possibility of component search, none of them are capable to highlight error-prone digitised symbols or sub-structures.

The paper is structured as follows. Section 2 explains the first tool, Section 3 explains the second one, and Section 4 concludes the paper.

[1] `https://www.geminivalve.com/best-piping-design-software/`
[2] `http://cfmgcomputing.blogspot.com/p/software-demos.html`

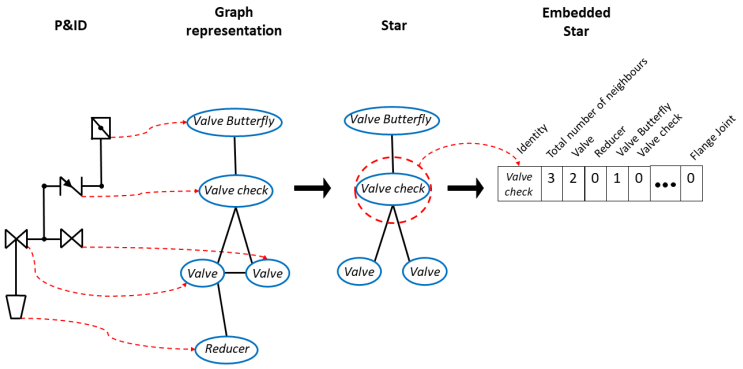

Figure 5: An example of embedding a *Valve check* star into a vector.

2 Predicting improperly identified components

The difference between the classical models (Figure 2) and our model (Figure 3) is the incorporation of the Automatic Validation module. The aim of this module is to deduce the identity of the components in the *Automatic CAD* and highlight the components that must be reviewed by an human expert, reducing in this way, the number of components that should be reviewed when any CAD document is generated by the Automatic Digitisation module. In the next two sub-sections, we detail the two main steps of this Automatic Validation module.

2.1 Graph representation and data embedding

The automatic validation method that we present is based on defining P&IDs as attributed graphs. In our graph, nodes represent components and edges represent pipelines that connect these components. Moreover, nodes have only one attribute, which is the component identity (valve, compressor,...) and edges are unattributed and unidirectional. In an attributed graph, a star (T_a) is defined as a local structure composed of a node, its connected edges and also the nodes that these edges connect (neighbour nodes). Our goal is to deduce the identity of each component given the set of pipelines connected to it and the components that connect these pipelines. For this reason we use the star, since by definition, this sub-structure contains this information.

Graphs have some limitations when they are applied to machine learning due to their intrinsic relational representation. This is because some trivial mathematical operations used in the traditional numeric machine learning representations have not an equivalence in the graph domain. Given an arbitrary set of graphs, a possible way to address this problem is to define an embedding function from the graph domain to a vector space [4]. Broadly speaking, an embedding function converts an attributed graph into a vector.

Since we want to use classical machine learning techniques to deduce component identities, we embed stars into vectors. Thus, each star is embedded in

a Euclidean space R^{n+2}, where n is the number of different component identities. The embedding of the i^{th} node in the graph (or the i^{th} component in the P&ID) is defined as a vector $E_i = (c_i, d_i, f_i^1, ..., f_i^n) \in R^{n+2}$ where c_i is the identity of the central node of the star; d_i is the number of edges in the star (or the number of connected pipelines to the central component); and f_i^p is the number of external nodes of the star that have the p identity, with $p = 1, ..., n$. A sample of a star embedding is presented in Figure 5. The output of this step is the set of all embedded stars in the graph representation of the CAD model.

2.2 Machine learning and verification

The Machine learning and verification step performs the following tasks:

- Firstly, each component in the P&ID represented by an embedded vector is introduced into the machine learning algorithm that returns the predicted component identity.
- Secondly, the identities of the components returned by the machine learning algorithm are contrasted with identities obtained from the digitised and contextualised *netlist* of the *Automatic CAD*. Note these components have not been verified by the engineer. Thus, this task discerns whether the deduced identities by our machine learning algorithm are the same or they are different from the CAD model.
- Thirdly, it detects the components of the *Automatic CAD* whose identities are different from the identities obtained by the machine learning algorithm. These detected components are highlighted to be validated by the human expert.

The learning set is composed of a CAD model, validated by a human expert, which has been embedded using our graph representation and data embedding step (Section 2.1). This CAD model must include a representative number of components per identity in order to assure the proper learning of the data. Usually, the larger the learning set is, the better the prediction given by the machine learning algorithm.

3 Searching groups of components

Before explaining the algorithm that implements the second tool, we set the following three premises that condition the values of its input parameters:

- Only engineers know how similar are two components in the P&ID. This knowledge is introduced into the system through the cost of substituting components imposed by the engineers before doing the query. For instance, if the engineer queries a graph that has a *valve* and wants to visualise all the groups of components similar to this query that have this *valve*, then they have to impose the substitution cost between a *valve*

and the rest of components to be infinite. Contrarily, if they know that two components are similar, then they can consider the substitution cost between them to be zero.

- In a similar way than the previous item, only engineers know how important is a component or a pipeline in the P&ID. This knowledge is considered in the node and edge deletion and insertion costs.
- Engineers want to visualise the locations where the query or similar queries appear in the P&ID. For this reason, it is desired that the method returns several connected components in the P&ID. These restrictions need to be handled by the search algorithm.

The rest of this section has been divided into two parts, in Section 3.1, the input and output parameters of our algorithm are defined and in Section 3.2 our algorithm is detailed.

3.1 Input and output parameters of our method

The **input** of our method is composed of:

- Q: A small graph that represents the query.
- G: A large graph that represents the P&ID.
- S_v: A square matrix of node substitution costs previously set by the engineer. Each cell is a non-negative real number that represents the cost of substituting two types of components. All the elements in the diagonal are zeros.
- D_v: A vector of node deletion costs previously set by the engineer. Each cell is a non-negative real number that depends on each specific component.
- D_e: Edge deletion cost. Since edges do not have attributes, the cost of deleting an edge is the same for all edges. It is a constant, D_e, previously set by the engineers.
- K: The number of compact sub-graphs the method has to return.

The **output** of the method is composed of:

- $\{f_1 : Q \rightarrow G$,..., $f_K : Q \rightarrow G\}$. A list of K node-to-node mappings between the query graph Q and the P&ID graph G.
- $\{D(Q, G, f_1)$,..., $D(Q, G, f_K)\}$. A list of K distances, given the query graph, the P&ID and the above mappings f_p, $1 \leq p \leq K$. If we define G_p as the nodes in G reached by substitutions in f_p, then $D(Q, G, f_p)$ is the distance between Q and G_p.

The returned list of mappings f_1,...,f_K hold the following four conditions:

- For each $1 \leq p \leq K$, the set of nodes $\{f_p(v)|v \in Q\} \subseteq G$ and their corresponding edges, defines a connected sub-graph.
- Components in P&ID can be reached by several mappings. Nevertheless, two mappings cannot be identical. Formally: $If\ f_p(v) = f_q(v), \forall v \in Q \Rightarrow p = q$.
- The mappings f_1,...,f_K are listed in ascending order on their distances. Formally: $D(Q, G, f_1) \leq D(Q, G, f_2) \leq$,..., $\leq D(Q, G, f_K)$.
- The mappings f_1,...,f_K might have to be the ones that return the minimum distance. Formally: $If\ f \notin \{f_1, ..., f_K\} \Rightarrow D(Q, G, f) \geqslant D(Q, G, f_p)$, $\forall p = 1, ..., K$.

3.2 Algorithm

The algorithm uses the sub-optimal error-tolerant graph matching algorithm *Belief Propagation* [13]. Its computational cost is only linear with respect to the number of nodes. It needs some initial node-to-node mappings, which are called *Seeds*, which in some applications could be a drawback but in our case, it is going to be crucial to generate several solutions.

Our algorithm has three main steps (Matlab implementation in [3]). In the first one, a cost matrix C is computed with dimensions $m \times n$, where m and n are the number of nodes of the query graph Q and the P&ID graph G, respectively. We assume, $m \leq n$. Each cell in C, $C(a, i)$, $1 \leq a \leq m$, $1 \leq i \leq n$, represents the cost of substituting the star T_a in Q by the star T_i in G.

In the second step, the K cells in the cost matrix that have the minimum value are selected. These substitution costs are used to set the K different *Seeds* that algorithm *Belief* is going to use in the K times it is run in the next step of our algorithm.

Finally, in the third step, the *Belief Propagation* algorithm [13] is executed K times. Each time, a different seed is used: $Seed_1$, ..., $Seed_K$. Using a different seed makes the algorithm to return a different mapping between the query Q and the P&ID G. This property could be considered a drawback in other methods but it becomes a must in our case.

Algorithm *Top-K-GED*
Input: Q, G, S_v, D_v, D_e, K
Output: f_1, ... , f_K, D_1, ... , D_K, being $D_p = D(Q, G, f_p)$, $p = 1, ..., K$
Begin Algorithm
$C = ComputeCostMatrix(Q, G, S_v, D_v, D_e)$
$(Seed_1, ..., Seed_K) = SelectLowerCostCells(C, K)$
For $p = 1, ..., K$
    $(f_p, D_p) = Belief(C, Seed_p)$
End For
End Algorithm

[3] http://deim.urv.cat/francesc.serratosa/SW/

4 Conclusions and future work

We have presented two tools to reduce the human effort while validating CAD documents that have been automatically generated from a class of complex engineering drawings called Pipping and Instrumentation diagrams (P&IDs).

The first tool detects incorrectly identified components in automatically generated CADs through learning their topology. To do so, we have represented the P&IDs by attributed graphs and we have embedded the local structures of components into vectors. Given each vector, a neural network has been used to predict the identity of the component represented by this vector. The second tool helps the engineer to search groups of connected components that are similar to a specific one. Its uniqueness is the fact that it returns several similar and compact sub-graphs. With the first tool we achieve an average reduction of approximately the 40% of the human effort, keeping an error-free process. With the second tool, we are able to find the queried group of components efficiently in more than the 80% of the cases, even achieving the 100% in some cases.

As a future work, we want to move our system from the laboratory to the industry, thus being in use in the digitisation process of P&ID sheets. These methods could be applied to other kind of industries in which the relational information between the components is available. We believe our methods could drastically reduce the human effort and therefore the economical and temporal cost of this essential task.

Acknowledgements

This project has received funding from Martí-Franquès Research Fellowship Programme of Universitat Rovira i Virgili, by The Data Lab and the Oil & Gas Innovation Centres (Scotland), and by Det Norske Veritas Germanischer Lloyd (DNV GL).

References

[1] Arroyo, E., Hoernicke, M., Rodríguez, P., Fay, A.: Automatic derivation of qualitative plant simulation models from legacy piping and instrumentation diagrams. Computers and Chemical Engineering **92**, 112–132 (2016). https://doi.org/10.1016/j.compchemeng.2016.04.040, `http://dx.doi.org/10.1016/j.compchemeng.2016.04.040`

[2] Cordella, L.P., Vento, M.: Symbol recognition in documents: a collection of techniques? International Journal on Document Analysis and Recognition **3**(2), 73–88 (2000)

[3] Elyan, E., Moreno-García, C.F., Jayne, C.: Symbols classification in engineering drawings. In: International Joint Conference on Neural Networks (IJCNN) (2018). https://doi.org/10.1109/IJCNN.2018.8489087

[4] Gibert, J., Valveny, E., Bunke, H.: Graph embedding in vector spaces by node attribute statistics. Pattern Recognition **45**(9), 3072–3083 (2012)

[5] Kang, S.O., Lee, E.B., Baek, H.K.: A digitization and conversion tool for imaged drawings to intelligent piping and instrumentation diagrams (P&ID). Energies **12**(2593), 1–26 (2019). https://doi.org/10.3390/en12132593

[6] Moreno-García, C.F., Elyan, E.: Digitisation of Assets from the Oil & Gas Industry: Challenges and Opportunities. In: International Conference on Document Analysis and Recognition (ICDAR). pp. 16–19. No. Workshop on Industrial Applications of Document Analysis and Recognition (WIADAR) (2019). https://doi.org/10.1109/ICDARW.2019.60122

[7] Moreno-García, C.F., Elyan, E., Jayne, C.: Heuristics-Based Detection to Improve Text / Graphics Segmentation in Complex Engineering Drawings. In: Engineering Applications of Neural Networks. vol. CCIS 744, pp. 87–98 (2017)

[8] Moreno-García, C.F., Elyan, E., Jayne, C.: New trends on digitisation of complex engineering drawings. Neural Computing and Applications **31**(6), 1695–1712 (2019). https://doi.org/10.1007/s00521-018-3583-1

[9] Rahul, R., Paliwal, S., Sharma, M., Vig, L.: Automatic Information Extraction from Piping and Instrumentation Diagrams. In: International Conference on Pattern Recognition Applications and Methods (ICPRAM). pp. 163–172 (2019). https://doi.org/10.5220/0007376401630172, `http://arxiv.org/abs/1901.11383`

[10] Rantala, M., Niemistö, H., Karhela, T., Sierla, S., Vyatkin, V.: Applying graph matching techniques to enhance reuse of plant design information. Computers in Industry **107**, 81–98 (2019). https://doi.org/10.1016/j.compind.2019.01.005, `https://doi.org/10.1016/j.compind.2019.01.005`

[11] Rica, E., Álvarez, S., Serratosa, F.: Group of components detection in engineering drawings based on graph matching. Engineering Applications of Artificial Intelligence **104**, 104404 (2021)

[12] Rica, E., Moreno-García, C.F., Álvarez, S., Serratosa, F.: Reducing human effort in engineering drawing validation. Computers in Industry **117**, 103198 (2020)

[13] Santacruz, P., Serratosa, F.: Error-tolerant graph matching in linear computational cost using an initial small partial matching. Pattern Recognition Letters (2018)

[14] Tombre, K., et al.: Graphics Recognition: Algorithms and Systems: Second International Workshop, GREC'97, Nancy, France, August 22-23, 1997, Selected Papers, vol. 2. Springer Science & Business Media (1998)